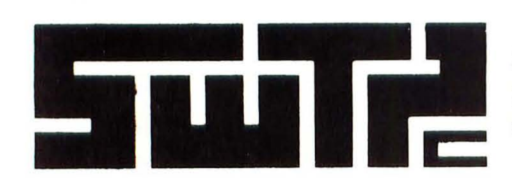

JANUARY 1979 Volume 4, Number 1 $2.00 in USA / $2.40 in Canada
BYTE
the small systems journal
ROBERT '78 TINNEY

Someday all terminals will be smart.......

- 128 Functions—software controlled
- 7 x 12 matrix, upper/lower case letters
- 50 to 38,400 baud—selectable
- 82 x 16 or 92 x 22 format—plus graphics
- Printer output port
- "CHERRY" keyboard

CT-82 Intelligent Terminal, assembled and tested. . . . . . .$795.00 ppd in Cont. U.S.



SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216

Circle 350 on inquiry card.



Model Z-2
Up to 512K of RAM/ROM



Model Z-2D
One or two disks
Up to 512K of RAM/ROM
Up to 184K of disk



System Two
Dual disk
Up to 512K of RAM/ROM
Up to 184K of disk

Fill your computer needs with the industry's most professional microcomputers

#1 IN RELIABILITY

When you choose Cromemco you get not only the industry's finest microcomputers but also the industry's widest microcomputer selection.

What's more, you get a computer from the manufacturer that computer dealers rate #1 in product reliability.*

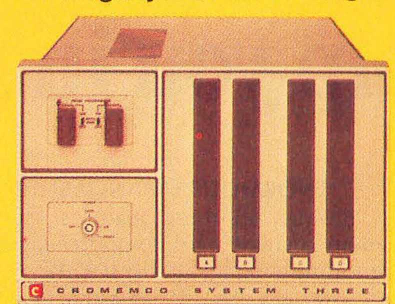
Your range of choice includes our advanced System Three with up to four 8″ disk drives. Or choose from the System Two and Z-2D with 5″ drives. Then for ROM-based work there's the Z2. Each of these computers further offers up to ½ megabyte of RAM (or ROM).

We say these are the industry's most professional microcomputers because they have outstanding features like these:

- **Z-80A microprocessor — operates at 250 nano second cycle time — nearly twice the speed of most others.**

*Rated in ***The 1977 Computer Store Survey*** by Image Resources, Westlake Village, CA.

Up to 512 kilobytes of RAM and 1 megabyte of disk storage



System Three
Two to four disks
Up to 512K of RAM/ROM
Up to 1 megabyte of disk

- **30-amp power supply — more than adequate for your most demanding application.**
- **21 card slots to allow for unparalleled system expansion using industry-standard S-100 cards.**
- **S-100 bus — don't overlook how important this is. It has the industry's widest support and Cromemco has professionally implemented it in a fully-shielded design.**
- **Cromemco card support of more than a dozen circuit cards for process control, business systems, and data acquisition including cards for A-D and D-A conversion, for interfacing daisy-wheel or dot-matrix printers, even a card for programming PROMs.**
- **The industry's most professional software support, including COBOL, FORTRAN IV, RATFOR, 16K Disk-Extended BASIC, Z-80 Macro Assembler, Cromemco Multi-User BASIC, Data Base Management System, Word Processing System — and more coming.**
- **Rugged, professional all-metal construction for rack (or bench or floor cabinet) mounting. Cabinets available.**

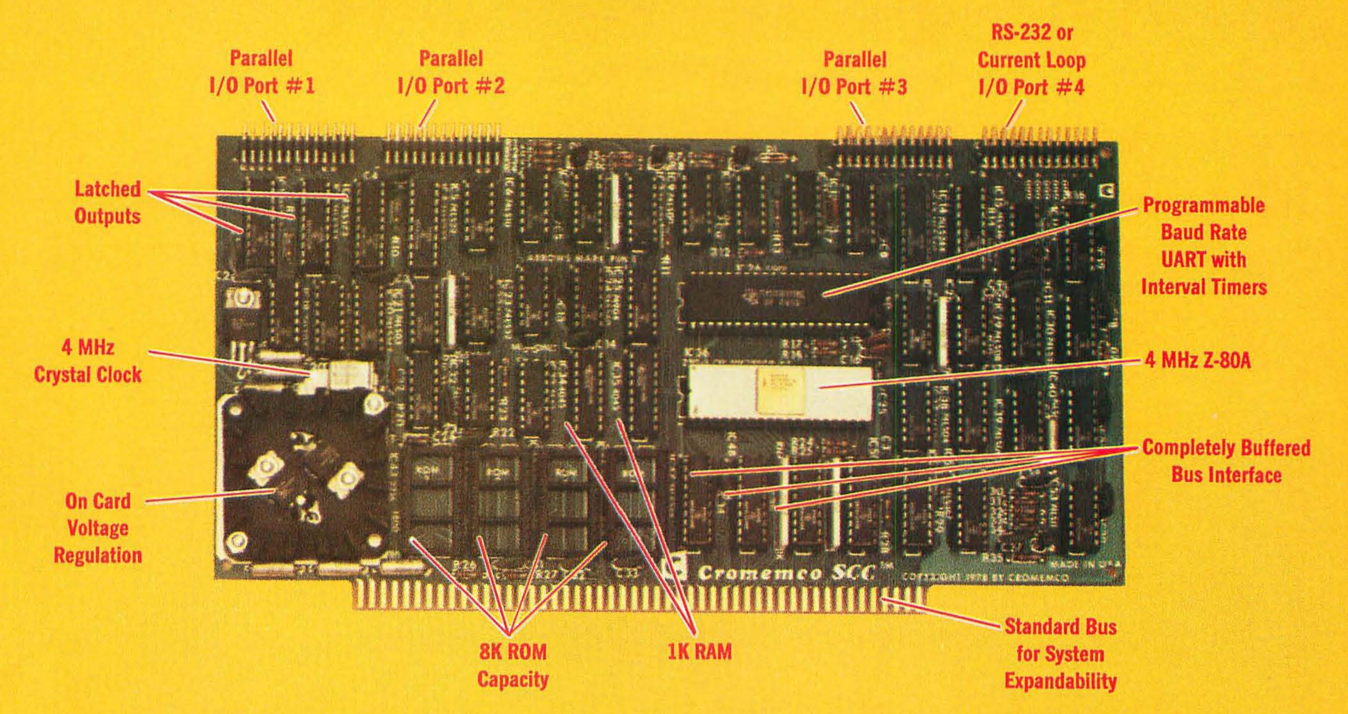
FOR TODAY AND TOMORROW

Cromemco computers will meet your needs now and in the future because of their unquestioned technical leadership, professionalism and enormous expandability.

See them today at your dealer. There's no substitute for getting the best.



Circle 80 on inquiry card.



The single card computer with the features that help you in real life

COMPLETE COMPUTER

In this advanced card you get a professional quality computer that meets today's engineering needs. And it's one that's complete. It lets you be up and running fast. All you need is a power supply and your ROM software.

The computer itself is super. Fast 4 MHz operation. Capacity for 8K bytes of ROM (uses 2716 PROMs which **can be programmed by our new 32K** BYTESAVER® PROM card). There's also 1K of on-board static RAM. Further, you get straightforward interfacing through an RS-232 serial interface with ultra-fast speed of up to 76,800 baud — software programmable.

Other features include 24 bits of bidirectional parallel I/O and five on-board programmable timers.

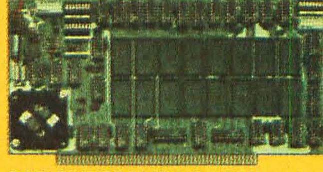
Add to that vectored interrupts.

ENORMOUS EXPANDABILITY

Besides all these features the Cromemco single card computer gives you enormous expandability if you ever need it. And it's easy to expand. First, you can expand with the new Cromemco 32K BYTESAVER PROM card mentioned above. Then there's Cromemco's broad line of S100-bus-compatible memory and I/O interface cards. Cards with features such as relay interface, analog interface, graphics interface, opto-isolator input, and A/D and D/A conversion. RAM and ROM cards, too.




Card Cage



32K BYTESAVER PROM card

EASY TO USE

Another convenience that makes the Model SCC computer easy to use is our Z-80 monitor and 3K Control BASIC (in two ROMs). With this optional software you're ready to go. The monitor gives you 12 commands. The BASIC, with 36 commands/functions, will directly access I/O ports and memory locations — and call machine language subroutines.

Finally, to simplify things to the ultimate, we even have convenient card cages. Rugged card cages. They hold cards firmly. No jiggling out of sockets.

AVAILABLE NOW/LOW PRICE

The Cromemco Model SCC is available now at a low price of only $450 factory assembled ($395 kit).

So act today. Get this high-capability computer working for you right away.

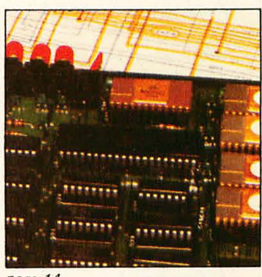


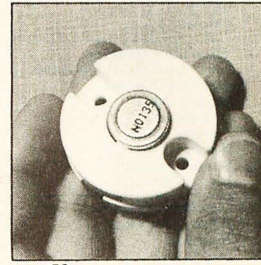
Circle 80 on inquiry card.

In the Queue

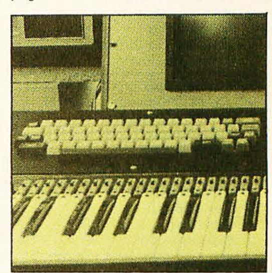
BYTE January 1979
Volume 4, Number 1

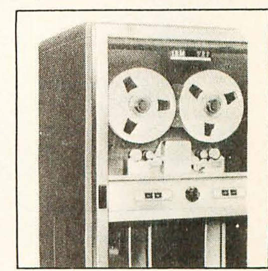
Foreground


page 14


page 56

Background


page 104


page 190

Nucleus



Cover Art: SCULPTING THE NEW TITANS *by Robert Tinney.*

BYTE is published monthly by BYTE Publications Inc, 70 Main St, Peterborough NH 03458. Address all mail except subscriptions to above address; phone (603) 924-7217. Address all editorial correspondence to the editor at the above address. Unacceptable manuscripts will be returned if accompanied by sufficient first class postage. Not responsible for lost manuscripts or photos. Opinions expressed by the authors are not necessarily those of BYTE. Address all subscriptions, change of address, Form 3579, and fulfillment complaints to BYTE Subscriptions, PO Box 590, Martinsville NJ 08836. Second class postage paid at Peterborough NH 03458 and at additional mailing offices—USPS Publication No. 102410. Subscriptions are $15 for one year, $27 for two years, and $39 for three years in the USA and its possessions. Add $5.50 per year for subscriptions to Canada and Mexico. $25 for a one year subscription by surface mail worldwide. Air delivery to selected areas at additional rates available upon request. $25 for a one year subscription by air delivery to Europe. Single copy price is $2.00 in the USA and its possessions, $2.40 in Canada and Mexico, $3.50 in Europe, and $4.00 elsewhere. Foreign subscriptions and sales should be remitted in United States funds. Printed in United States of America.

Subscription WATS Line: (800) 258-5485

This month's cover is entitled *Sculpting the New Titans*, a fantasy by Robert Tinney that emphasizes the new crop of innovative circuitry being chipped out of silicon by microcomputer engineers. Of course, in real life the tools are more likely to be a computer, plotter, and wafer fabrication facility.

Among the new devices is Motorola's new 6809 processor, expected to be available in the next few months. Successor to the 6800, the 6809 is an 8 bit processor that can perform 16 bit operations. The two chief architects of the 6809, Terry Ritter and Joel Boney, begin a 3 part discussion of the new circuit in **A Microprocessor for the Revolution: The 6809, Part 1: Design Philosophy.** *page 14*

In this BYTE

This month Steve Ciarcia begins the 3 part series, **Build a Computer Controlled Security System for Your Home.** Part 1 covers everything you need to keep your home or business secure, from the right types of sensors to detailed software examples. *page 56*

Of the four standard languages of the American National Standards Institute BASIC, FORTRAN and PL/I are familiar. Less so is MUMPS, used for many business and science applications. Get **An Exposure to MUMPS** in David Sherertz' article. *page 74*

Implementing a mailing list on your computer is a practical application for many personal computer experimenters. If you have a floppy disk based system, read **A Computerized Mailing List** by Thomas E Doyle and find out how easy it is to do. *page 84*

Last month David Buckingham described interesting and intricate patterns that occur in John H Conway's game of Life. This month we take a look at some of the methods and schemes for implementing a Life program on your computer. Mark D Niemiec reveals a Life enthusiast's trade secrets in **Life Algorithms.** *page 90*

In **The Digicast System** A I Halsema describes a new technique for receiving computer programs and information broadcast over the subcarrier frequency bands of FM radio stations. Similar systems are now in operation in Europe. *page 100*

One of the problems in creating computer synthesized music is that many systems allow only one note at a time to be entered—a time-consuming restriction. Steven Roberts describes a way around the dilemma in **Polyphony Made Easy.** *page 104*

Computers cannot play perfect chess, but sometimes a computer program can do surprisingly well. The human United States chess champion discovered this fact while competing against the world champion computer chess program. That confrontation was recorded by J R Douglas in **Grandmaster Walter Browne versus Chess 4.6.** *page 110*

BNF (Backus Normal Form or Backus-Naur Form) is a standardized method for abbreviating certain statements made about programming languages. Used extensively in books and articles dealing with compiler and interpreter design, it is explained for those new to the subject in W D Maurer's article, **An Introduction to BNF.** *page 116*

In **Creating a Chess Player, Part 4: Strategy in Computer Chess,** authors Peter Frey and Larry Atkin describe some clever optimization techniques that can significantly reduce the amount of computer time needed to evaluate end game positions. *page 126*

In this issue's Languages Forum: **IPS, An Unorthodox High Level Language,** Karl Meinzer presents an introduction to his creation, a machine independent high level language interpreter presently implemented for the RCA 1802 and 8080 processors. *page 146*

Test your computer vocabulary by puzzling over Gwen Hadley's story of **GOTOlocks and the Three Sorts.** *page 174*

In **Elements of Statistical Computation,** Alan Forsythe covers some of the dos and don'ts of determining the mean and standard deviation of a set of data. *page 182*

Logic probes are handy for troubleshooting digital logic. Read about James L Woodward's circuit in **An Audible Logic Test Probe.** It plays two tones through a speaker corresponding to high or low logic levels, and it costs very little. *page 186*

We tend to take the microprocessors of today for granted, forgetting the power hungry days of the 1950s. In his article, **History of Computers: The IBM 704,** Keith Reid-Green looks at one of the last vacuum tube computers to be commercially marketed. *page 190*

"My Shugart followed me home."

"After working all day with the computer at work, it's a kick to get down to Basic at home. And one thing that makes it more fun is my Shugart minifloppy™. We use Shugart drives at work, so when I bought my own system I made sure it had a minifloppy drive.

"Why? Shugart invented the minifloppy. The guys who designed our system at work tell me that Shugart is the leader in floppy design and has more drives in use than any other manufacturer. If Shugart drives are reliable enough for hard-working business computers, they've got to be a good value for my home system.

"When I'm working on my programs late at night, I can't wait for cassette storage. My minifloppy gives me fast random access and data transfer. The little minidiskettes™ store plenty of data and file easily too.

"I made the right decision when I bought a system with the minifloppy. when you lay out your own hard-earned cash, you want reliability and performance. Do what I did. Get a system with the minifloppy."

If it isn't Shugart, it isn't minifloppy.



435 Oakmead Parkway, Sunnyvale, California 94086

For a list of manufacturers featuring Shugart's minifloppy in their systems, circle reader response number.

Circle 320 on inquiry card.

TM minifloppy is a registered trademark of Shugart Associates

Editorial

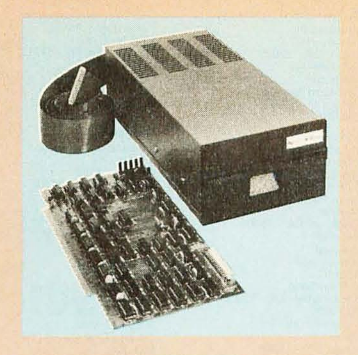
A Short Note on Pascal Progress and Other Topics

by Carl Helmers

As our January issue deadline approached, we received a memo from Ken Bowles of the University of California at San Diego, concerning yet another development in the progress of Pascal as a language appropriate for small machines. This is a plan (expected to reach fruition in early 1979) for making the UCSD Pascal system available on the Apple II computer. The result, of course, is a computer with dual floppy disks and Pascal system capability that, sold through a computer store, would be priced at about $3000. Of course, the UCSD people are most overjoyed by this because it makes available for educational uses a Pascal compatible machine which is unmatched in price. This Apple II version of UCSD Pascal will be supported with one 12 K byte read only memory card and main memory of 48 K bytes. It is expected that this configuration will set a standard for small systems.

Following the Western Digital Pascal Micro Engine computer noted last month, we now have Pascal compatible systems at every level of design: from the chip level to that of the integrated consumer computers. Once again, our enthusiasm for Pascal must be reiterated: it is widely available in a standard form and quite compatible with personal computers.

But the languages of the computer culture do not end with BASIC, Pascal and APL—all used or discussed in these pages to some extent. Languages for computer use is a topic too broad to be served by just a few examples. Other languages worth exploring in these pages might include such well known examples as SNOBOL, GPM, LISP, and various low level symbolic Polish notation interpreters such as Karl Meinzer's IPS interpreter which is discussed in this issue. It may be only a matter of time before we have systems with such languages built in. After all, the electronics engineering is more or less identical for all small computers, and to distinguish them from one another in the marketplace, such "trivial" matters of taste as language and operating systems conventions may soon be important criteria for the computer buyer at the local computer store showroom. There is plenty of room for different languages and operating system conventions as the market grows larger. Such diversity of choice is one of the joys of the present era in computing.■



Specifications:
S-100 compatible. MFM encoding, 35 tracks with ten 512-byte sectors per track. 179,200 bytes on double density SA-400 and North Star BASIC, DOS, and Monitor included.

For further information, write for full color catalog or contact your local computer store.

New from North Star

Double Density Performance at Single Density Prices

The new HORIZON computer and Micro Disk System now record in double density! That means each new Shugart SA-400 minifloppy disk drive accesses 180K bytes of on-line information. All double density HORIZON computers and Micro Disk Systems have a redesigned controller which allows the use of quadruple capacity disk drives as they become available in early 1979. A three-drive North Star System with quadruple capacity disk drives will access over a megabyte of on-line information. But, best of all there's no price increase for double density models.

North Star BASIC and DOS have been upgraded to accommodate the increased capacity and yet run existing programs with little or no change. The new disk system also supports single density, so existing single density diskettes can still be used. Single density SA-400 drives previously purchased with North Star systems can also be used.

Pricing

HORIZON with one double density SA-400 minifloppy (180K bytes), 16K RAM, Z80A processor and serial I/O port: $1599 kit, $1899 assembled.

MICRO DISK SYSTEM with one double density SA-400 minifloppy, controller board and power regulation: $699 kit, $799 assembled. (Cabinet and power supply $39 extra each.)

NORTH STAR ★ COMPUTERS
2547 Ninth Street
Berkeley, California 94710
(415) 549-0858

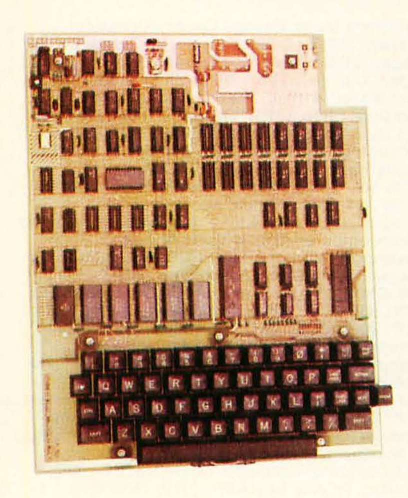
The Age of Affordable Personal Computing Has Finally Arrived.

Ohio Scientific has made a major breakthrough in small computer technology which dramatically reduces the cost of personal computers. By use of custom LSI micro circuits, we have managed to put a complete ultra high performance computer and all necessary interfaces, including the keyboard and power supply, on a single printed circuit board. This new computer actually has more features and higher performance than some home or personal computers that are selling today for up to $2000. It is more powerful than computer systems which cost over $20,000 in the early 1970's.

This new machine can entertain your whole family with spectacular video games and cartoons, made possible by its ultra high resolution graphics and super fast BASIC. It can help you with your personal finances and budget planning, made possible by its decimal arithmetic ability and cassette data storage capabilities. It can assist you in school or industry as an ultra powerful scientific calculator, made possible by its advanced scientific math functions and built-in "immediate" mode which allows complex problem solving without programming! This computer can actually entertain your children while it educates them in topics ranging from naming the Presidents of the United States to tutoring trigonometry all possible by its fast extended BASIC, graphics and data storage ability.

The machine can be economically expanded to assist in your business, remotely control your home, communicate with other computers and perform many other tasks via the broadest line of expansion accessories in the microcomputer industry.

This machine is super easy to use because it communicates naturally in BASIC, an English-like programming language. So you can easily instruct it or program it to do whatever you want, *but you don't have to.* You don't because it comes with a complete software library on cassette including programs for each application stated above. Ohio Scientific also offers you hundreds of inexpensive programs on ready-to-run cassettes. Program it yourself or just enjoy it; the choice is yours.



Ohio Scientific offers you this remarkable new computer two ways.

Challenger 1P $349
Fully packaged with power supply. Just plug in a video monitor or TV through an RF converter to be up and running.



Superboard II $279
For electronic buffs. Fully assembled and tested. Requires +5V. at 3 Amps and a video monitor or TV with RF converter to be up and running.

Standard Features

- Uses the ultra powerful 6502 microprocessor
- 8K Microsoft BASIC-in-ROM
  Full feature BASIC runs faster than currently available personal computers and all 8080-based business computers.
- 4K static RAM on board expandable to 8K
- Full 53-key keyboard with upper/lower case and user programmability
- Kansas City standard audio cassette interface for high reliability
- Full machine code monitor and I/O utilities in ROM
- Direct access video display has 1K of dedicated memory (besides 4K user memory), features upper case, lower case, graphics and gaming characters for an effective screen resolution of up to 256 by 256 points. Normal TV's with overscan display about 24 rows of 24 characters; without overscan up to 30 X 30 characters.

Extras

- Available expander board features 24K static RAM (additional), dual mini-floppy interface, port adapter for printer and modem and an OSI 48 line expansion interface.
- Assembler/editor and extended machine code monitor available.

ORDER FORM

Order direct or from your local Ohio Scientific dealer.

☐ I'm interested. Send me information on your:
☐ Personal Computers ☐ Business Systems
☐ Send me a Superboard II $279 enclosed
☐ Send me a Challenger 1P $349 enclosed
☐ Include 4 more K of RAM (8K Total) $69 more enclosed

Name ______

Address ______

City ______ State ______ Zip ______

Payment by: BAC (VISA) ____ Master Charge ____ Money Order ____

Credit Card Account # ______

Expires ______ Interbank #(Master Charge) ______

Ohio Residents add 4% Sales Tax

TOTAL CHARGED OR ENCLOSED ______

All orders shipped insured UPS unless otherwise requested. FOB Aurora, OH

Interested in a bigger system? Ohio Scientific offers 15 other models of microcomputer systems ranging from single board units to 74 million byte hard disk systems.

OHIO SCIENTIFIC

America's Largest Full Line Microcomputer Company

1333 S. Chillicothe Road • Aurora, Ohio 44202 (216) 562-3101

Circle 290 on inquiry card.

Letters

COMPUTER INSURANCE?

In response to Leonard Anderson's letter (October 1978 BYTE, page 157) regarding insurance for computers, I think a few comments are in order:

1. A standard homeowners policy *will* extend to cover a microcomputer system. As Mr Anderson mentioned, this type of policy is generally set at minimum amount of coverage for personal property within a dwelling at 50 percent of the dwelling itself. However, this is a minimum amount and it can be increased usually at a rate of about $2 per thousand, to whatever limit is necessary to cover the contents involved.
2. Coverage under a homeowners policy, however, provides coverage only on a "named peril" basis. That is, such types of losses as fire, smoke, vandalism and theft would be covered. On the other hand, losses such as breakage would not be covered under this type of policy.
3. If a broader form of coverage is needed, it should be possible to have a separate personal property floater written to cover the microcomputer. We have done this in some instances and it does give an "all risk" form of coverage. It is necessary, however, to specify the values involved in your system when writing this type of policy.
4. In any event, both the homeowners and the personal property floater cover only the value of raw media, eg: disks and tapes will be covered only for the value of the disks or tapes, not for the value of the programs entered on those media. This would not apply, however, to commercially purchased disks and tapes which already have programs on them, since the insurable value would be the purchase price.
5. In all the above instances, we have assumed that the computer is a personal computer and is not being used for business purposes. If the system *is* being used for business purposes, it is necessary to write a separate policy to cover this system. A homeowners policy will specifically exclude equipment used for business purposes, and in that event I would recommend that you write either a personal property floater or, for those persons who have an office but take their computer home for additional programming, etc, it would probably be possible to extend the business policies to cover the computer.
6. For those persons involved in fairly extensive business use of computers, there are speciality policies written by a limited number of insurance companies, (eg: St Paul Insurance Company and Fireman's Fund) which are specifically designed to cover both computer systems and all forms of media.

I hope these comments will be of some help to your readers.

Bob Henderson CPCU, CLU
Hendersons Inc Insurance Brokers
6548 S Bright Av
Whittier CA 90601

SOME FOOTNOTES FROM JAPAN

I mentioned "two recent Tokyo shows" in my letter *Some Notes From Japan* in September 1978 BYTE, page 17. The Sharp PC1300 appeared at the business show, as mentioned; the other items (Mitsubishi, Panafacom, NEC, etc) were at the microcomputer show. Sorry about the omission.

A conversion error of 680 grams to 8 ounces appeared in my letter. 1 pound 8 ounces is nearer.

Also, NEC's Level 2 BASIC has only just appeared (September 1978). It's a replacement read only memory for the 8080 system.

K S Wilkinson
CPO Box 1748
Tokyo JAPAN 100-91

JUMPING BACKWARDS WITH WADUZITDO

I found the WADUZITDO program (September 1978 BYTE, page 166) fun to play with and I noticed that the JUMP function could easily be modified to permit backward jumps, as follows:

```
00A5 FE 0100   LDX    LOC
00A8 C4 0F     AND B  #$0F
00AA 26 02     BNE    JF
```

Now, WADUZITDO can jump back in an absolute fashion (rather than rela-

Continued on page 166



Sol.® The small computer that won't fence you in.

A lot of semantic nonsense is being tossed around by some of the makers of so-called "personal" computers. To hear them tell it, an investment of a few hundred dollars will give you a computer to run your small business, do financial planning, analyze data in the engineering or scientific lab — and when day is done play games by the hour.

Well, the game part is true. The rest of the claims should be taken with a grain of salt. Only a few personal computers have the capacity to grow and handle meaningful work in a very real sense. And they don't come for peanuts.

Remember, there's no free lunch.

So before you buy any personal computer, consider Sol.® It costs more at the start but less in the end. It can grow with your ability to use it. Sol is not cheap. But it's not a delusion either.

Sol small computers are at the very top of the microcomputer spectrum. They stand up to the capabilities of mini systems costing four times as much.

No wonder we call it the serious solution to the small computer question.

Sol is the small computer system to do the general ledger and the payroll. Solve engineering and scientific problems. Use it for word processing. Program it for computer aided instruction. Use it anywhere you want versatile computer power!

Build computer power with our software.

At Processor Technology we've tailored a group of high-level languages, an assembler and other packages to suit the wide capabilities of our hardware.

Our exclusive Extended BASIC is a fine example. This BASIC features complete matrix functions. It comes on cassette or in a disk version which has random as well as sequential files.

Processor Technology FORTRAN is similar to FORTRAN IV and has a full set of extensions designed for the "stand alone" computer environment.

Our PILOT is an excellent text oriented language for teachers.

Sold and serviced only by the best dealers.

Sol Systems are sold and serviced by an outstanding group of conveniently located computer stores throughout the U.S. and Canada.

For more information contact your nearest dealer in the adjacent list. Or write Department B, Processor Technology, 7100 Johnson Industrial Drive, Pleasanton, CA 94566. Phone (415) 829-2600.

In sum, all small computers are not created equal and Sol users know it to their everlasting satisfaction.

Processor Technology


DYNABYTE DB8/2
POWER ON
RESET
POWER OFF
HALT
DYNABYTE DB8/1
POWER ON
RESET
POWER OFF
HALT
DYNABYTE DB8/4
POWER ON
POWER OFF

A Microprocessor for the Revolution: The 6809

Part 1: Design Philosophy

Terry Ritter
Joel Boney
Motorola Inc
3501 Ed Bluestein Blvd
Austin TX 78721

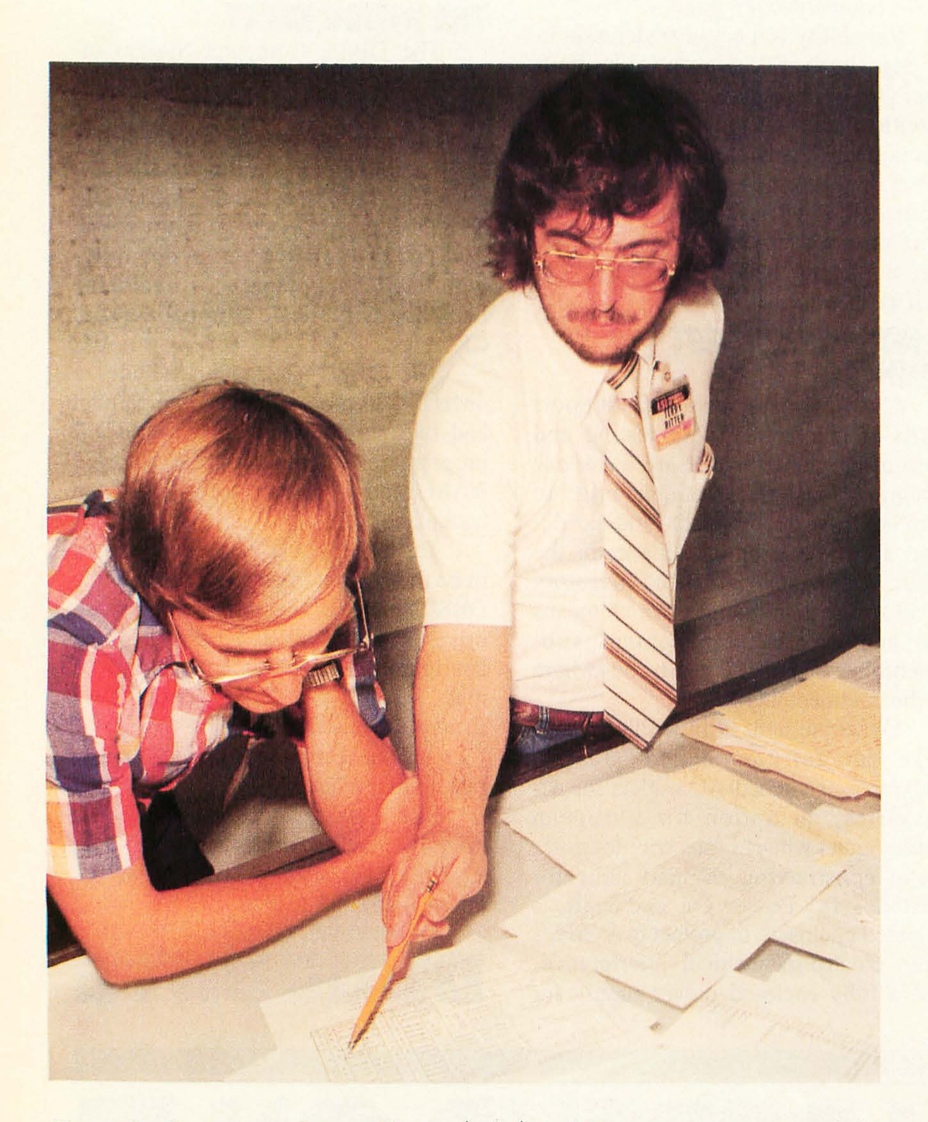


Photo 1: Systems architects Ritter (right) and Boney review some of the 6809 design documents. This work results in a complete description of the desired part in a 200 page design specification. The specification is then used by logic designers to develop flowcharts of internal operations on a cycle by cycle basis.

This is a story. It is a story of computers in general, specifically microcomputers, and of one particular microprocessor — with revolutionary social change lurking in the background. The story could well be imaginary, but it happens to be true. In this 3 part series we will describe the design of what we feel is the best 8 bit machine so far made by human: the Motorola M6809.

Philosophy

A new day is breaking; after a long slow twilight of design the sun is beginning to rise on the microprocessor revolution. For the first time we have mass production computers; expensive, custom, cottage industry designs take on less importance.

Microprocessors are real computers. The first and second generation devices are not very sophisticated as processors go, but they *are* general-purpose logic machines. Any microprocessor can eventually be made to solve the same problems as any large scale computer, although this may be an easier or harder task depending on the microprocessor. (Naturally, some jobs require doing processing *fast*, in real time. We are not discussing those right now. We *are* discussing getting a big job done *sometime.*) What differentiates the classes is a hierarchy of technology, size, performance, and, curiously, philosophy of use.

A processor of given capability has a fixed general complexity in terms of digital logic elements. Consider the computers that were built using the first solid state technology. In short, they consisted of many thousands of individual transistors and other parts on hundreds of different printed circuit cards using thousands of connections and miles of connecting wire. A big computer was a big project and a very big expense. This simple economic fact fossilized a whole generation of technology into the "big computer philosophy."

Because the big computer was so expensive, time on the computer was regarded as a limited and therefore valuable resource. Certainly the time was valuable to researchers who could now look more deeply into their equations than ever before. Computer time was valuable to business people who became at least marginally capable of analyzing the performance of an unwieldy

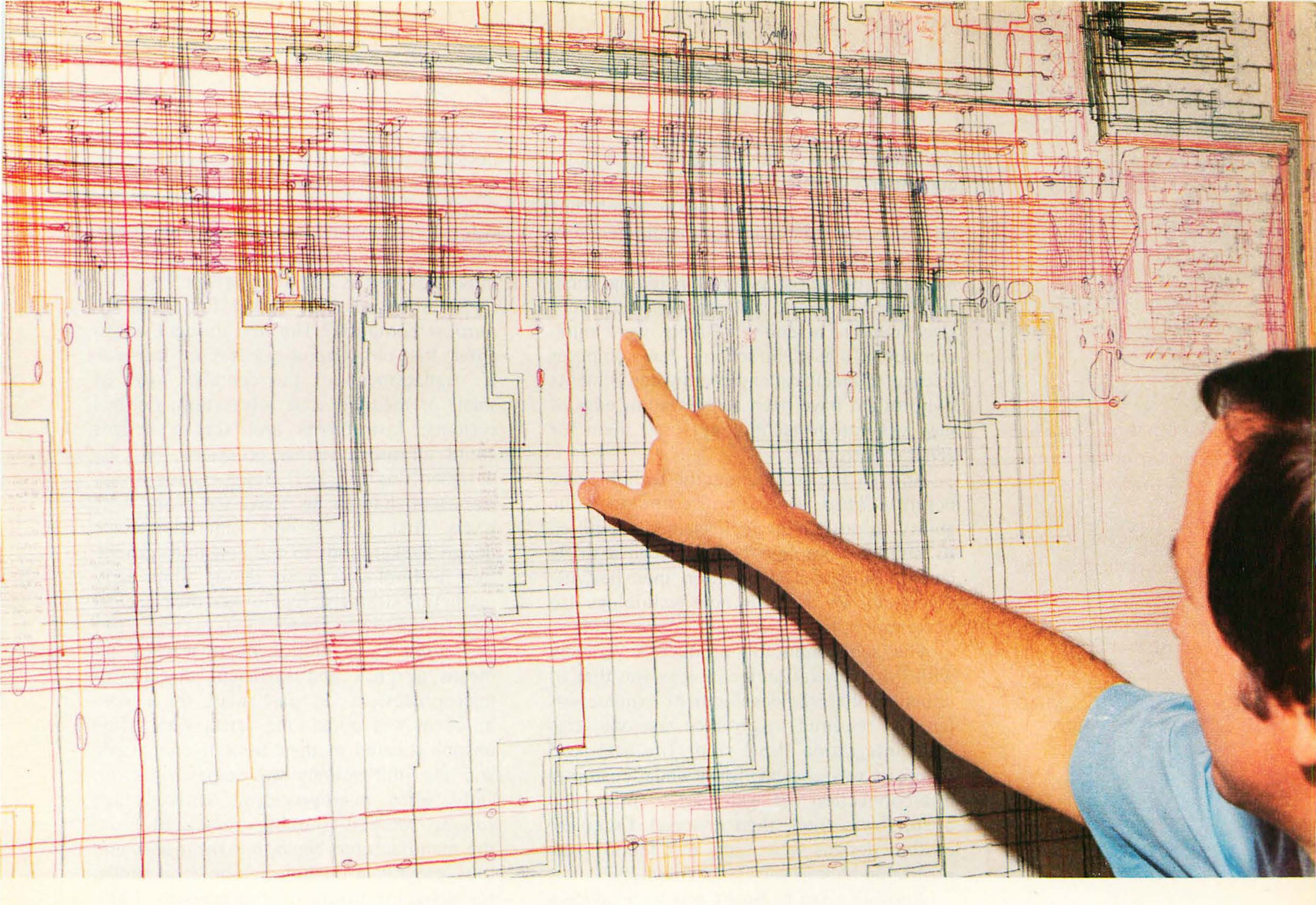


Photo 2: 6809 logic design. Design engineer Wayne Harrington inspects a portion of 6809's processor logic blueprint at the Motorola Austin plant. The print is colored by systems engineers to partition the logic for the logic-equivalent TTL "breadboard."

bureaucratic organization. And the computer makers clearly thought that processor time was valuable too; it was a severely limited resource, worth *as much as the market would bear.*

Processor time *was* a limited resource. But some of us, a few small groups of technologists, are about to change that situation. And we hope we will also change how people look at computers, and how professionals see them too. Computer time should be cheap; people time is 70 years and counting down.

The large computer, being a very expensive resource, quickly justified the capital required to investigate optimum use of that resource. Among the principal results of these projects was the development of batch mode multiprocessing. The computer itself would save up the various tasks it had to do, then change from one to the other at computer speeds. This minimized the wasted time between jobs and spawned the concept of an *operating system*.

People were in the position of waiting for the computer, not because they were less important than the machine, but precisely because it *was* a limited resource (the problems it solved were not).

Electronics know-how continued to develop, producing second generation solid state technology: families of digital logic integrated circuits replaced discrete transistor designs. This new technology was exploited in two main thrusts: big computers could be made conceptually bigger (or faster, or better) for the same expense, or

About the Authors

Joel Boney and Terry Ritter are with the Motorola 6800 Microprocessor Design Group in Austin TX. Joel is responsible for the software input into the design of the 6800 family processors and peripheral parts and was a co-architect of the M6809. Terry Ritter is a microcomponent architect, responsible for specification of the 6809 advanced microprocessor. While with Motorola, Terry has been co-architect of the 6809, and co-architect as well of the 6847 and 68047 video display generator integrated circuits. He holds a BSES from the University of Texas at Austin and Joel Boney has a BSE from the University of South Florida.

The other major device needed for home computers — the video display generator color TV interface — is presently in volume production. Several versions are available, many derived from the original Motorola architecture.

computers could be made physically smaller and less expensive. These new, smaller computers (minicomputers) filled market segments which could afford a sizable but not huge investment in both equipment and expertise. Laboratories could use them, for example. But most people, including scientists and engineers, still used only the very large central machines. Rarely were minicomputers placed in schools; few computer science or electrical engineering departments (who might have been at the leading edge of new generation technology) used them for general instruction.

And so the semiconductor technologists began a *third* generation technology: the ability to put all the logic elements required to build a complete computer on a single chip of silicon. The question then became, "How do we use this new technology (to make money)?"

The semiconductor producer's problem with third generation technology was that an unbelievably large development expense was (and is) required to produce just one large scale integration (LSI) chip. The best road to profit was unclear; for a while, customer interconnection of gate array integrated circuits was tried, then dropped. Complete custom designs were (and are) found to be profitable only in very large volumes.

Another road to profit was to produce a few *programmable* large scale integration devices which could satisfy the market needs (in terms of large quantities of different systems) and the factory's needs (in terms of volume production of exactly the same device). Naturally, the general-purpose computer was seen as a possible answer.

So what was the market for a general-purpose computer? The first thought was to enter the old second generation markets; ie: replacement of the complex logic of small or medium scale integration. Control systems, instruments and special designs could all use a similar processor, but the designer was the key. Designers (or design managers) had to be converted from their heavy first and second generation logic design backgrounds to the new third generation technology. In so doing, some early marketing strategists overlooked the principal microprocessor markets.

Random logic replacement was by no means a quick and sufficient market for microprocessors. In particular, the design-in cycle was quite long, users were often unsophisticated in their uses of computers, and the unit volume was somewhat small. Only when microprocessors entered high volume markets (hobby, games, etc) did the manufacturers begin to make money and thus provide a credible reason (and funds) for designing future microprocessors. Naturally, the users who wanted more features were surprised that it was taking so long to get new designs — they *knew* what was needed.

Thus semiconductor makers began to realize that their market was more oriented to hobby applications than to logic replacement, and was more generalized than they had thought. But even the hobby market was saturable.

Meanwhile companies continued to improve production and reduce costs, and competition drove prices into the ground. Where could they sell enough computers for real volume production, they wondered. One answer was the personal computer!

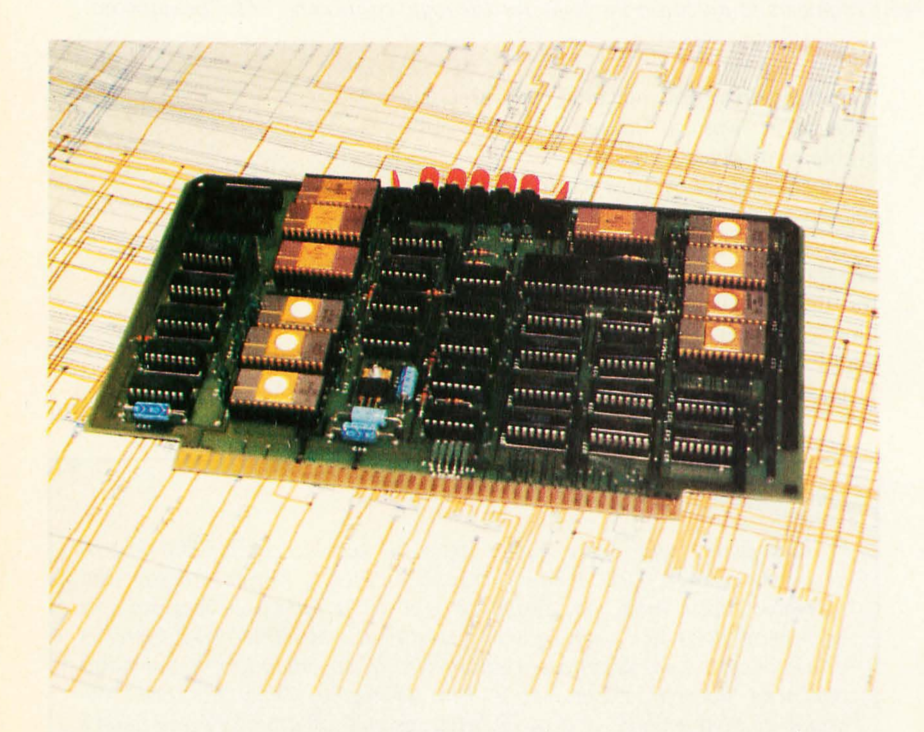


Photo 3: 6809 emulator board. Software and systems engineers implement a functional equivalent of the 6809 as a 6800 program. A 6800 to 6809 cross assembler allows 6809 programs to be assembled and then executed as a check of the architectural design.

Design of Large Scale Integration Parts

The design of a complex large scale integration (LSI) part may be conveniently broken into three phases: the architectural design, the logic and circuit design/architectural review, and the layout software and hardware (breadboard) simulations. Each phase has its own requirements.

The architect/systems designers represent the use of the device, the needs of the marketplace and the future needs of all customers. They propose what a specific cus-

tomer should have that could also be used by other customers, possibly in different ways. They advocate what the customers will really want, even if no customers can be identified who know that they will want it, that it is possible or that it is inexpensive. The attitude that "I know what is best for you" may be irritating to most people, but it is necessary in order to make maximum use of a limited resource (in this case, a single LSI design). The architect eventually generates the design specification used in subsequent phases of the design.

Logic design consists of the production of a cycle by cycle flowchart and the derivation of the equations and logic circuitry necessary to implement the specified design. This is a job of immense complexity and detail, but it is absolutely crucial to the entire project. Throughout this phase, the specification may be iterated toward a local optimum of maximum features at minimum logic (and thus, cost). The architectural design continues, and techniques are developed to cross-check on the logical correctness of the architecture.

The third phase is the most hectic in terms of demands and involvement. By this time, many people know what the product is and see the resulting part merely as the turning of an implementation "crank." It seems to those who are not involved in this phase that more effort could cause that crank to turn faster. Since the product could be sold immediately, delay is seen as a real loss of income. *In actual practice*, more effort will sometimes "break the crank."

A medium scale integration logic implementation (usually transistor-transistor logic, for speed) is required to verify the logic design. A processor emulation may require ten different boards of 80 medium scale integrated circuits each and hundreds of board to board interconnections. Each board will likely require separate testing, and only then will the emulation represent the processor to come. Extensive test programs are required to check out each facet of the part, each instruction, and each addressing mode. This testing may detect logic design errors that will have to be fixed at all levels of design.

Circuit design, in the context of the semiconductor industry, depends upon running computer simulations (which require sophisticated device models) of signals at various nodes to verify that they will meet the necessary speed requirement. Transistors are sized and polysilicon lines changed to provide reliable worst case operation.

Layout is the actual task of arranging transistors and interconnections to implement the logic diagram. Circuit design results will indicate appropriate transistor sizes and polysilicon widths; these must now be arranged for minimum area. Every attempt is made to make general logic "cells" which can be used in many places across the integrated circuit, but area minimization is the principal concern.

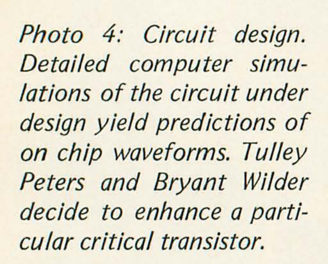


Photo 4: Circuit design. Detailed computer simulations of the circuit under design yield predictions of on chip waveforms. Tulley Peters and Bryant Wilder decide to enhance a particular critical transistor.

The layout for the chip eventually exists only as a computer data base. Each cell is individually digitized into the computer, where it can be arbitrarily positioned, modified or replicated as desired. Large 2 by 3 m (6.5 by 10 feet) plots of various areas of the chip are hand checked to the logic diagram by layout and circuit designers as final checks of the implemented circuit.

When layout is complete, the computer data base that represents the chip design is sent to the *mask* shop (the mask is a photographic stencil of the part used in the manufacturing process). At the mask shop precision plotting and photographic step and repeat techniques are used to produce glass plates for each mask layer. Each mask covers an entire wafer with etched nickel or chrome layouts at real chip size. (A typical LSI device will be between 5 by 5 and 7.5 by 7.5 mm (0.2 by 0.2 and 0.3 by 0.3 inches). These masks are used to expose photosensitive etch resist that will protect some areas of the wafer from the chemical processes which selectively add the impurities that create transistors.

Actual processing steps are quite similar for each part. But the processing itself is a variable, and it will not be known until final testing exactly how many parts will turn out to be saleable. Therefore, a best estimate is taken, and the required number of wafers (of a particular device) is started and processed. The whole industry revolves around highly trained production engineers, chemists and others who process wafers to highly secret recipes. Some recipes work, some don't. You find out which ones do by testing.

Each die (ie: individual large scale integration circuit) is tested while still on the wafer; failing devices are marked with a blob of ink. The wafer is sawed into individual dies and the good devices placed into a plastic or ceramic package base. The connection pads are "die bonded" to the exposed internal lead frame with very tiny wire. The package is then sealed and tested again.

Testing a device having only 40 pins but which has up to 40,000 internal transistors is no mean trick nor a minor expense. Furthermore, the device must execute all operations properly at the worst case system conditions (which may be high or low extremes of temperature, voltage and loading) and work with other devices on a common bus. Thus, the device is not specified to its own maximum operating speed, but rather the speed of a worst case system. Motorola microprocessors can usually be made to run much faster (and much slower) than their guaranteed worst case specifications.

Photo 5: Checking the flowcharts. Logic and circuit designer Bryant Wilder compares the specification to one of the flowcharts. The flowcharts are used to develop Boolean equations for the required logic; those equations are then used to generate a logic diagram.

Project Goals

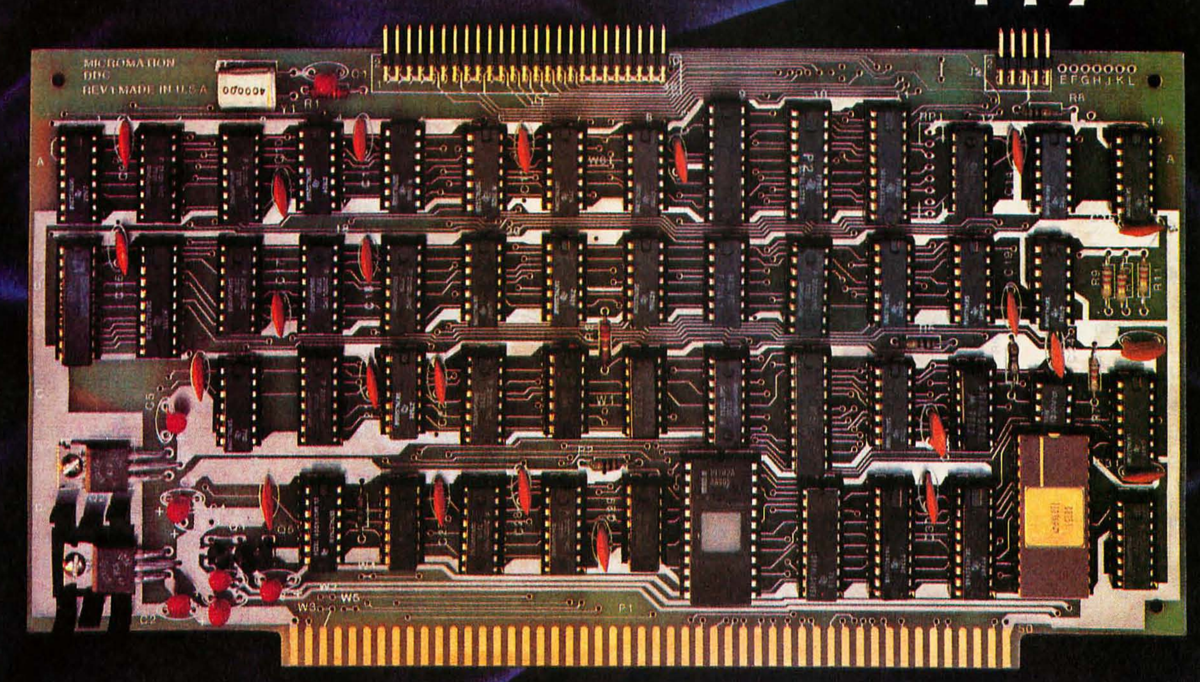
The 6809 project started life with a number of (mostly unformalized) goals. The principal public goal was to upgrade the 6800 processor to be definitely superior to the 8 bit competition. (The Motorola 68000 project will address the 16 bit market with what we believe will be another superior processor.) Many people, including many customers, felt that all that had to be done was to add another index register (Y), a few supporting instructions (LDY, STY) and correct some of the past omissions (PSHX, PULX, PSHY, PULY). Since this would mean a rather complete redesign anyway, it made little sense to stop there.

A more philosophical goal – thus one much less useful in discussions with engineers and managers (who had their own opinions of what the project should be) – was to minimize software costs. This led to an extensive, and thus hard to explain, sequence of logic that went somewhat like this:

Q: *How do we reduce software costs?*

A: 1. Write code in a block structured high level language.

Micromation has done for the S-100 bus what IBM did for the floppy disk.



Reliably doubled capacity.

Double Capacity
The DOUBLER — Micromation's latest advance in floppy disk technology — doubles the capacity of floppy disk systems. Over 500 KBytes are recorded on each side of an 8" disk. This means bigger files for more powerful systems.

Double Speed
Data transfer with the DOUBLER is twice as fast — 500 Kbits per second. And since there is twice as much data on each track, your drive steps only half as much — so your system runs faster than it ever has before!

Increased Reliability
That's right — even better reliability. Why? Because we did it the IBM way. IBM designed 2D formatting — so it has to be reliable. Micromation's innovative, state-of-the-art design incorporates write precompensation electronics and a phase lock oscillator on a single, all digital, S-100 circuit board. So we guarantee the DOUBLER will be more dependable than your present single density controller — and we warantee the DOUBLER for a full year.

Unbeatable Convenience
It couldn't be easier to step up to double density. The DOUBLER operates automatically in either single or double density. Just insert a diskette and you're running properly. You can transfer files between single or double density diskettes without any software or hardware changes — or even operate with one single and one double density diskette.

Installation is a snap. There's a hardware UART on board and the software is all ready to go. An onboard 2708 EPROM contains the bootstrap. There's even jump-on-reset circuitry so you can operate without a front panel. And, of course, we include utilities to format diskettes.

Universally Versatile
The DOUBLER will operate with all industry-standard mini and full-sized drives. And it will work in any 8080 or Z-80 S-100 computer operating at 2 to 4 MHz. The DOUBLER will support up to four double or single headed drives.

Fully Compatible
The DOUBLER is compatible with CP/M* version 1.4. If you have a CP/M* 1.4 system, just add our CBIOS — or you can buy our ready-to-boot version. Install the new controller, connect any terminal to the RS-232 interface, and boot off your new double-sized, double-speed system. You still can use all your old software without any changes.

Completely Affordable
All Micromation products are fully assembled, thoroughly tested, include complete documentation, and are priced for value:

DOUBLER double density controller	**$ 495.**
MEGABOX dual drive double density system	**2,295.**
ZEPHER — Per Sci double density system	**2,595.**
Z-PLUS — MEGABOX 32 KZ-80 computer	**4,295.**

Available
The DOUBLER is available NOW at your local computer store.

Micromation Inc. 524 Union Street San Francisco California 94133 / 415 398-0289



Where there's always more in store.

*CP/M is a trademark of Digital Research.

Circle 223 on inquiry card.

For your SWTP 6800 Computer . . .

PERCOM's™ FLOPPY DISK SYSTEM the LFD-400

Ready to plug in and run the moment you receive it. Nothing else to buy, no extra memory. No "booting" with PerCom MINIDOS™, the remarkable disk operating system on EPROM. Expandable to either two or three drives. Outstanding operating, utility and application programs.



For the low $599.95 price, you not only get the disk drive, drive power supply, SS-50 bus controller/interface card, and MINIDOS™, you also receive:
• an attractive metal enclosure • a fully assembled and tested interconnecting cable • a 70-page instruction manual that includes operating instructions, schematics, service procedures and a complete listing of MINIDOS™ • technical memo updates — helpful hints which supplement the manual instructions • a 90-day limited warranty.

SOFTWARE FOR THE LFD-400 SYSTEM

Disk operating and file management systems

INDEX™ The most advanced disk operating and file management system available for the 6800. INterrupt Driven EXecutive operating system features file-and-device-independent, queue-buffered character stream I/O. Linked-file disk architecture, with automatic file creation and allocation for ASCII and binary files, supports sequential and semi-random access disk files. Multi-level file name directory includes name, extension, version, protection and date. Requires 8K RAM at $A000. Diskette includes numerous utilities $99.95

MINIDOS-PLUSX An easy-to-use DOS for the small computing system. Supports up to 31 named files. Available on ROM or diskette complete with source listing $39.95

BASIC Interpreters and Compilers

SUPER BASIC A 10K extended disk BASIC interpreter for the 6800. Faster than SWTP BASIC. Handles data files. Programs may be prepared using a text editor described below $49.95

BASIC BANDAID™ Turn SWTP 8K BASIC into a random access data file disk BASIC. Includes many speed improvements, and program disk CHAINing $17.95

STRUBAL+™ A STRUctured BAsic Language compiler for the professional programmer. 14-digit floating point, strings, scientific functions, 2-dimensional arrays. Requires 16K RAM and Linkage Editor (see below). Use one of the following text editors to prepare programs. Complete with RUN-TIME and FLOATING POINT packages $249.95

Text Editors and Processors

EDIT68 Hemenway Associates' powerful disk-based text editor. May be used to create programs and data files. Supports MACROS which perform complex, repetitive editing functions. Permits text files larger than available RAM to be created and edited $39.95

TOUCHUP™ Modifies TSC's Text Editor and Text Processor for PerCom disk operation. ROLL function permits text files larger than available RAM to be created and edited. Supplied on diskette complete with source listing $17.95

Assemblers

PerCom 6800 SYMBOLIC ASSEMBLER Specify assembly options at time of assembly with this symbolic assembler. Source listing on diskette $29.95

MACRO-RELOCATING ASSEMBLER Hemenway Associates' assembler for the programming professional. Generates relocatable linking object code. Supports MACROS. Permits conditional assembly $79.95

LINKAGE EDITOR — for STRUBAL+™ and the MACRO-Relocating assembler $49.95

CROSS REFERENCE Utility program that produces a cross-reference listing of an input source listing file $29.95

Business Applications

GENERAL LEDGER SYSTEM Accommodates up to 250 accounts. Financial information immediately available — no sorting required. Audit trail information permits tracking from GL record data back to source document. User defines account numbers $199.95

FULL FUNCTION MAILING LIST 700 addresses per diskette. Powerful search, sort, create and update capability $99.95

PERCOM FINDER™ General purpose information retrieval system and data base manager $99.95

™ trademark of PERCOM Data Company, Inc.

Ordering information

To order, call toll free *1-800-527-1592*. MC and VISA welcome. COD orders require 30% deposit plus 5% handling charge. Allow three weeks for delivery. Allow three extra weeks if payment is by personal check. Texas residents add 5% sales tax.

PERCOM *'peripherals for personal computing'*

Circle 301 on inquiry card.

2. Distribute the code in mass production read only memories.

Q: *Why aren't many read only memories being used now?*

A: 1. The great opportunities for error in assembly language allow many mistakes which incur severe read only memory costs.
   2. The present architecture is not suitable for read only memories.

Q: *In what way are the second generation processors unsuitable?*

A: It is very difficult to use a read only memory in any other context than that for which it was originally developed. It is hard to use the same read only memory on systems built by different vendors. Simply having different input and output (IO) or using a different memory location is usually enough to make the read only memory product useless.

Q: *What is needed?*

A: 1. Position independent code.
   2. Temporary variables on the stack.
   3. Indirect operations through the stack for input and output.
   4. Absolute indirect operations for system branch tables.

And so it went. How could we make a device that would answer the software problems of two generations of processors? How, indeed!

Design Decisions

Usually an engineering project may be pursued in many ways, but only one way at a time. The ever present hope is that this one time will be the only time necessary. Furthermore, it would be nice to get the project over with as soon as possible to get on with selling some products. (A rapid return on investment is especially important in a time of rapid inflation.) To these honorable ends certain decisions are made which delineate the investment and risk undertaken in an attempt to achieve a new product.

The 6809 project was no exception. To minimize project risk it was decided that the 6809 would be built on the same technological base as the recently completed 6800 depletion load redesign. In particular, the machine would be a random logic computer with essentially dynamic internal operation. It would use the reliable 6800 type of storage register. Functions would be limited to those befitting a producible sized device.

The 6809 part would have to be compatible with the defined 6800 bus and 6800 peripherals. This decision would extend the life of parts already in production and minimize testing peripheral devices for a particular processor (6800 *versus* 6809). Bus compatibility doesn't have to mean identity – the new device could have considerably improved specifications but could not do worse than the specifications for the existing device. This mandate was a little tricky when you consider that we were dealing with a more complex device using exactly the same technology, but there was a slight edge: the advancing very large scale integration (VLSI) learning curve.

One wide range decision was that the new device would be an improved 6800 part. The widely known 6800 architecture would be iterated and improved, but no radical departure would be considered. In fact, the new device should be code compatible with the 6800 at some level.

Compatibility was the basis for the 6809 architectural design. It implied that the 6809 could capitalize on the extensive familiarity with the 6800. 6800 programmers could be programming for the 6809 almost immediately and could learn and use new addressing modes and features as they were needed. This decision also ended any consideration of a radically new architecture for the machine before it was begun.

A corporation selling into a given market is necessarily limited to moderate innovation. Any vast product change requires reeducation of both the internal marketing organization *and* the customer base before mass sales can proceed. Consequently, designers have to restrict their creativity to conform to the market desires. The amount of change actually implemented, produced and seen by society is the true meaning of a computer "generation." In the end, society itself defines the limits of a new generation, and a design years ahead of its time may well fail in the marketplace.

M6800 Data Analysis

Once the initial philosophical and marketing trade-offs were made, construction of the final form of the M6809 began. By this time a large number of M6800 programs had been written by both Motorola and our customers, so it was felt that a good place to start design of the 6809 was to analyze large amounts of existing 6800 source code. Surprisingly, the data gathered about 6800 usage of instructions and addressing modes agreed substantially with similar data previously compiled for minicomputers and maxicomputers. By far the most common instructions were the loads and stores, which accounted for over 38 percent of all 6800 instructions. Next were the subroutine calls



MICROPRO INTERNATIONAL CORPORATION

MicroPro International Corporation™

professional quality you can count on!

proudly announces

SUPER-SORT©

The ultimate in high performance sort/merge

Specifications

□High Performance Tournament/Heapsort Algorithm
□Multiple Input Files Read in Parallel for True Merge Operation. Up to 32 Sort and 32 Merge Input Files Handled.
□Dynamically Invokable User Exit Routines
□Record Selection via SELECT/EXCLUDE Statements
□Handles Fixed *and* Variable Length Records
□Handles Fixed *and* Variable Length Fields
□Handles up to 32 Sort Keys with Intermixed Sequence Indicators, Data Types and Alternate Collating Sequences
□Handles Multi-Volume Diskette Files
□Compatible with CP/M* and any Derivative Including ADOS, IMDOS, CDOS, etc.
□8080/8085/Z-80 Compatible
□Specialized Optimizations for Floppy Disk Environment
□Keyword Command Input for Easy Operator Entry
□Benchmarked at Over 560 Records per Minute!
□Invokable as a Subroutine from FORTRAN, COBOL and Assembler
□Furnished in Relocatable and Executable Form for Easy Load-Address Definition
□Optional TAGSORT Operation
□Data Types Include ASCII, EBCDIC, Binary, BCD (COBOL Packed Decimal), etc.
□Supports CP/M-compatible Diskette Files under BASIC, FORTRAN, COBOL and Assembler.

Price $250** includes manual and single densitydiskette. Manual only $15 refundable with purchase.

WORD-MASTER©

The last word in text editing

Specifications

□Compatible with any "Dumb" CRT Possessing Addressable Cursor and Backspace (Includes Hazeltine - All Models, SOROC, Lear Siegler, IMSAI-VIOC, ADDS Regent, Behive, etc.)
□Bi-Directional Word Tab, Line Tab, Screen Tab
□Bi-Directional Word Delete, Line Delete and Character Delete
□Quad-Directional Cursor Movements
□Mid-Line Insert and Delete
□Automatic RAM/Diskette Buffering With No User Intervention
□Nested Command Looping with Conditional Execution
□Global String Search, Global String Replace
□Scratch-Pad Buffer for Text Movement, Global Replication, and String Command Storage
□Multiple Input File Merging with User-Controlled Insertions for Easy Document Assembly
□Multiple Output File Control by Section Under User Control
□Compatible with CP/M and Most of its Derivatives Including CDOS, IMDOS, etc.
□Use TEX* for Print Formatting Functions

Price $150** includes manual and single density diskette. Manual only $9.00 refundable with purchase.

*CP/M and TEX are Trademarks of Digital Research.

** Prices and Specifications subject to change without notice.

Dealer Inquiries Invited: Call (Northern California) (707) 544-2865, (415) 398-7062, (209) 445-0511, (408) 279-8980, (916) 485-7619 (Southern California) (213) 224-1619, (714) 634-2908. Outside California Call Collect (707) 544-2865. Principal offices located at 5810 Commerce Blvd., Rohnert Park, CA 94928

Circle 224 on inquiry card.

Instruction Class	Percent Usage
Loads	23.4
Stores	15.3
Subroutine calls and returns	13.0
Conditional branches	11.0
Unconditional branches and jumps	6.5
Compares and tests	6.2
Increments and decrements	6.1
Clear	4.4
Adds and subtracts	2.8
All others	11.3

Table 1: 6800 instruction types based on static analysis of 25,000 lines of 6800 source code. In static analysis the actual number of occurrences of each instruction is tallied from program listings. In the alternate technique, called dynamic analysis, the number of occurrences of an instruction is tallied while the program is running. An instruction inside a program loop would therefore be counted more than once.

and returns with 13 percent, conditional branches with 11 percent and unconditional jumps and branches with 6.5 percent (see table 1). Neither the arithmetic nor logical instructions had as high a usage as might have been expected. Clearly then, enhancements that would improve the utility and power of the data movements (such as load and stores) would yield the largest return on investment, followed by improvements to subroutine linkage and parameter passing.

Further analysis indicated that the number of load and store index register instructions (16 bits) was too large to be attributable solely to index register manipulation or even to the lack of a second index register. This information, combined with a relatively high ratio between straight adds or subtracts and adds with carry and subtracts with borrow, indicated that quite a few simple 16 bit operations were being performed on existing 6800s.

It was therefore felt the M6809 must support the most common 16 bit operations on the accumulators and index registers.

Perhaps the most interesting data was that which pertained to addressing modes. The six major 6800 addressing modes (Direct, Extended, Immediate, Indexed, Relative, Accumulator) had nearly equal usage, which indicated that programmers actually took advantage of the bytes to be saved by direct (page zero) addressing and indexed addressing. Furthermore the offsets for indexed instructions showed that 93 percent of the offsets were either 0 or less than 32 (see table 2).

This information was used to greatly expand the addressing modes (as discussed later) without making the 6800 programs require more code when converted to run on the 6809. Also the number of increment or decrement index register instructions in loops indicated that autoincrementing and autodecrementing would be beneficial. Autodecrementing and autoincrementing are similar to indexing except the index register used is decremented before, or incremented after, the addressing operation takes place.

As all programmers and even architects like ourselves eventually learn, consistent and uniform instruction sets are used more effectively than instruction sets that treat similar resources (IO, registers or data) in dissimilar ways. For example, the least used instructions on the 6800 were those that dealt with the A accumulator in specific ways that did not apply to the B accumulator (eg: ABA: add B to A, CBA: compare B to A). It's not that these instructions are not useful, it's just that programmers will not use inconsistent instructions or addressing modes. Consistency became the battle cry of the M6809 designers!

Customer Inputs

At the completion of the 6800 analysis stage, the first preliminary design specification for the 6809 was generated. This preliminary specification was then taken to about 30 customers who represented a cross section of current 6800 users, as well as some customers and consultants known to be hostile to the 6800. With these customer visits we hoped to resolve two major questions about the 6809's architecture:

1) Which architecture was more desirable, 8 bit or 16 bit?
2) Did 6809 compatibility with the 6800 need to occur at the object level or at the source level?

Most customers felt that an 8 bit architecture was adequate for their upcoming applications, and they did not want to pay the price penalty for 16 bits as long as the 6809 included the most common 16 bit operations such as add, subtract, load, store, com-

Table 2: Size of offsets used in 6800 indexed addressing, based on static analysis of 25,000 lines of 6800 source code.

Index Offset	Percent Usage
0	40.0
1–31	53.0
32–63	1.0
64–255	6.0


North Star Systems Go To Work
More and more, you see the North Star HORIZON computer at work: in business, research, and education. Its high performance qualifies the HORIZON for demanding professional applications. Over 10,000 users during the past two years have proven that North Star hardware has the reliability for day-in, day-out computing. The HORIZON is now a serious candidate for any small system installation.
SOFTWARE IS THE KEY TO HORIZON MATURITY
North Star BASIC and DOS have been used to develop hundreds of commercial program packages. These packages establish that North Star software has the completeness and convenience necessary for serious program development. Because of the many independent vendors offering software using North Star BASIC and DOS, the HORIZON owner now has the widest selection of software in the microcomputer industry! Software available includes: word processing, general ledger, accounts payable/receivable, mailing list processing, inventory and income tax preparation. Program development systems such as assemblers, debuggers, editors, PILOT and FORTRAN are also available.
EXPAND YOUR HORIZON
The basic HORIZON computer includes a Z80 microprocessor, 16K bytes of RAM memory, an I/O interface and one Shugart minifloppy disk drive. The HORIZON can be expanded to 60K bytes or more of RAM, three disk drives, and three I/O interfaces. Performance can be enhanced by the addition of the North Star hardware floating point board. Also, S-100 bus products from other manufacturers may be used to expand the HORIZON.
For more information, contact your local computer store.
North Star Computers
HORIZON
NORTH ★ STAR COMPUTERS
2547 Ninth Street
Berkeley, California 94710
(415) 549-0858
Circle 285 on inquiry card.

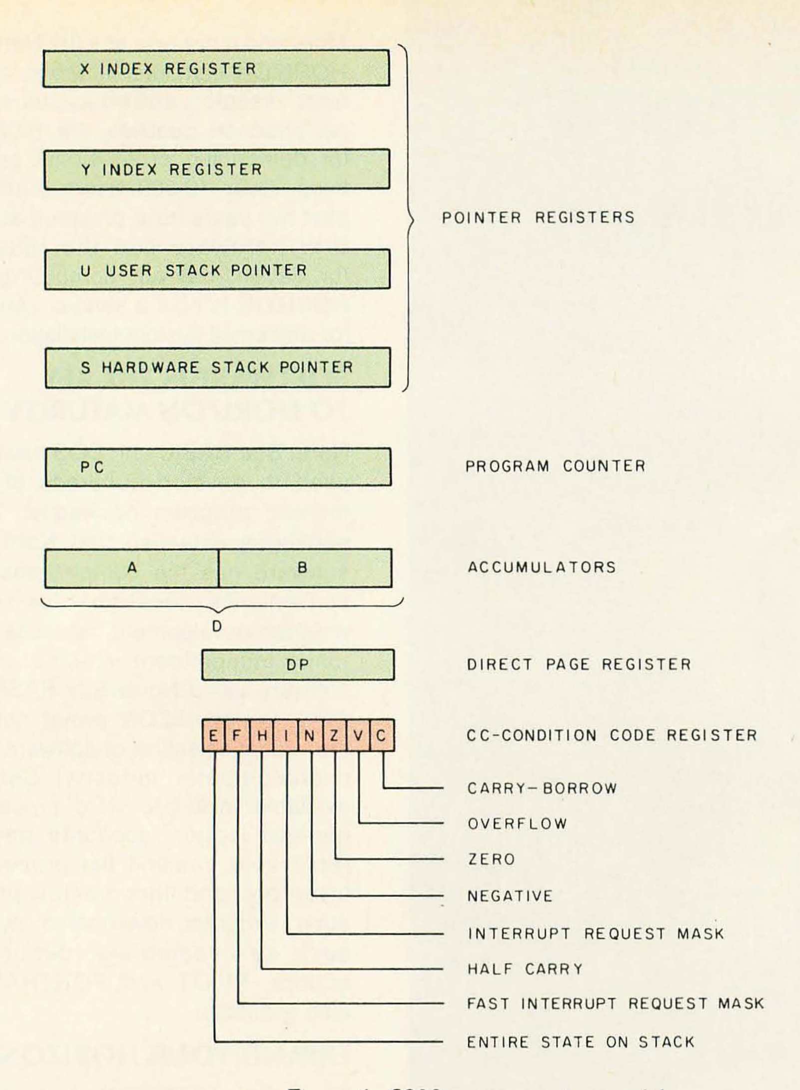


Figure 1: 6809 programming model.

pare and multiply. Many were interested, though, in Motorola's advanced 16 bit processor (68000) for future 16 bit applications. From the very inception of the 6809 project it was a requirement that the 6809 would be compatible with the 6800. Whether this compatibility needed to occur at the object code level or at the assembly language (source code) level was a question we felt our customers should help us answer. Virtually every customer indicated that source compatibility was sufficient because they would not try to use 6800 read only memories in 6809 systems. Most customers indicated that they would take advantage of the 6800 compatibility in order to initially convert running 6800 programs into running 6809 programs, and then modify the 6809 code to take advantage of the 6809's features.

The decision not to be object code compatible was an easy one for us since it meant that we could remap the 6800 op codes in a manner guaranteed to produce more byte efficient and faster 6809 programs. The remapping of op codes was greatly affected by the 6800 data analyses. Some low occurrence 6800 instructions were combined into consistent 2 byte instructions, allowing the more useful instructions to take fewer bytes and execute faster. Also, some 6800 instructions were eliminated completely in favor of 2 instruction sequences. These sequences are generated automatically by our 6809 assembler when the 6800 mnemonic is recognized. This remapping in favor of more often used functions results in 6809 programs that require only one half to two thirds as much memory as 6800 programs, and run faster.

M6809 Registers

What, then, are the pertinent features that make the 6809 a next generation processor? In the following paragraphs we will attempt to highlight the improvements made to the 6800. The programming model for the 6809 (figure 1) consists of four 8 bit registers and five 16 bit registers.

The A and B accumulators are the same as those of the 6800 except that they can also be catenated into the A:B pair, called the D register, for 16 bit operations.

The condition codes are similar to the 6800, with the inclusion of two new bits. The F bit is the interrupt mask bit for the new fast interrupt. The fast interrupt (FIRQ) only stacks the program counter and condition code register when an interrupt occurs. The interrupt routine is then responsible for stacking any registers it uses. The E bit is set when the registers are stacked during interrupts if the entire register set was saved (as in nonmaskable and maskable interrupts) or cleared if the short register set was saved (for a fast interrupt).

On the 6800, an instruction with direct mode (or page zero) addressing consisted of an op code followed by an 8 bit value that defined the lower eight bits of an address. The upper eight bits were always assumed to be zero. Thus, direct addressing could only address locations in the lowest 256 bytes of memory. The 6809 adds versatility to this addressing mode by defining an 8 bit direct page register that defines the upper eight bits of address for all direct addressing instructions. This allows direct mode addressing to be used throughout the entire address space of the machine. To maintain 6800 compatibility, the direct page register is set to 0 on reset.

Four 16 bit indexable registers are included in the 6809. They are the X, Y, U and S registers. The X register is the familiar 6800 index register, and the S register is the hardware stack pointer. The Y register is a

Type	Forms	Nonindirect				Indirect			
		Source	Post Byte	+ ~	+ #	Source	Post Byte	+ ~	+ #
Constant offset from R	no offset	,R	1RR00100	0	0	[,R]	1RR10100	3	0
	5 bit offset	n,R	0RRnnnnn	1	0		defaults to 8-bit		
	8 bit offset	n,R	1RR01000	1	1	[n,R]	1RR11000	4	1
	16 bit offset	n,R	1RR01001	4	2	[n,R]	1RR11001	7	2
Accumulator offset from R	A register offset	A,R	1RR00110	1	0	[A,R]	1RR10110	4	0
	B register offset	B,R	1RR00101	1	0	[B,R]	1RR10101	4	0
	D register offset	D,R	1RR01011	4	0	[D,R]	1RR11011	7	0
Autoincrement/ –decrement R	increment by 1	,R+	1RR00000	2	0		not allowed		
	increment by 2	,R++	1RR00001	3	0	[,R++]	1RR10001	6	0
	decrement by 1	,–R	1RR00010	2	0		not allowed		
	decrement by 2	,– –R	1RR00011	3	0	[,– –R]	1RR10011	6	0
Constant offset from program counter	8 bit offset	n, PCR	1XX01100	1	1	[n, PCR]	1XX11100	4	1
	16 bit offset	n, PCR	1XX01101	5	2	[n, PCR]	1XX11101	8	2
Extended		use nonindexed				[n]	10011111	5	2

Table 3: Indexed addressing modes. All instructions with indexed addressing have a base size and number of cycles. The + ~ and + # columns indicate the number of additional cycles and bytes for the particular variation. The post byte op code is the byte that immediately follows the normal op code.

second index register; the U register is the user stack pointer. All four registers can be used in all indexing operations and the U and S registers are also stack pointers. The S register is used during interrupts and subroutine calls by the hardware to stack return addresses and machine state.

The last 16 bit register is the program counter. In certain 6809 addressing modes, the program counter can also be used as an index register to achieve position independent code.

Addressing Modes

It was our opinion that the best way to improve an existing architecture and maintain source compatibility was to add powerful addressing modes. In our view, the 6809 has the most powerful addressing modes available on any microprocessor. Powerful addressing modes helped us achieve our goals of position independence, reentrancy, recursion, consistency and easy implementation of block structured high level languages.

All the 6800 addressing modes (Immediate, Extended, Direct, Indexed, Accumulator, Relative and Inherent) are supported on the 6809 with the direct mode of addressing made more useful by the inclusion of the direct page register (DPR).

The direct page register usage and direct addressing need some explanation, since they can be very effective when used correctly. For example, since global variables are referenced frequently in high level language execution, the direct page register can be used to point to a page containing the global variables while the stack contains the local variables, which are also referenced frequently. This creates very efficient code which is safe since the compiler keeps track of the direct page register. The direct page register can also be used effectively and safely in a multitasking environment where the real time operating system allocates a difference base page for each task.

On the other hand, it would be quite dangerous to indiscriminately reallocate the direct page register frequently, such as within subroutines or loops, since it might become very easy to lose track of the current direct page register value. Therefore, even though the direct page register is unstructured, we included it because, when used correctly, the byte savings are significant. Also, to make direct addressing more useful, the read modify write instructions on the 6809 now have all memory addressing modes: Direct, Extended and Indexed.

The major improvements in the 6809's addressing modes were made by greatly expanding the indexed addressing modes as well as making all indexable instructions applicable to the X, Y, U and S registers (see table 3).

Indexed addressing with an offset is familiar to 6800 users, but the 6809 allows the offset to be any of four possible lengths: 0, 5, 8 or 16 bits, and the offsets are signed two's complement values. This allows greater flexibility in addressing while achieving maximum byte efficiency. The inclusion of the 16 bit offset allows the role of index register and offset to be reversed if desired. A further enhancement allows all of the above modes

```
                              NAM    AUTOEX
                              OPT    LLEN=80
                       *
                       ********************************************
                       *   COMPARE STRINGS SUB
                       *
                       *   FIND AN INPUT ASCII STRING POINTED TO BY THE
                       *   X-REGISTER IN A TEXT BUFFER POINTED TO BY THE
                       *   Y-REGISTER. THE BUFFER IS TERMINATED BY A
                       *   BYTE CONTAINING A NEGATIVE VALUE. ON ENTRY
                       *   A CONTAINS THE LENGTH OF THE INPUT STRING. ON
                       *   EXIT, Y CONTAINS THE POINTER TO THE START
                       *   OF THE MATCHED STRING + 1 IFF Z IS SET. IFF Z
                       *   IS NOT SET THE INPUT STRING WAS NOT FOUND.
                       *
                       *   ENTRY:
                       *      X POINTS TO INPUT STRING
                       *      Y POINTS TO TEXT BUFFER
                       *      A LENGTH OF INPUT STRING
                       *   EXIT:
                       *      IFF Z=1 THEN Y POINTS TO MATCHED STRING + 1
                       *      IFF Z=0 THEN NO MATCH
                       *      X IS DESTROYED
                       *      B IS DESTROYED
                       *
                       ********************************************
                       *
0100                          ORG    $100
0100  E6 A0     6      CMPSTR LDB    ,Y+        GET BUFFER CHARACTER
0102  2A 01     3             BPL    CMP1       BRANCH IF NOT AT BUFFER END
0104  39        5             RTS               NO MATCH, Z=0
0105  E1 84     4      CMP1   CMPB   ,X         COMPARE TO FIRST STRING CHAR.
0107  26 F7     3             BNE    CMPSTR     BRANCH ON NO COMPARE
                       * SAVE STATE SO SEARCH CAN BE RESUMED IF IT FAILS
0109  34 32     9             PSHS   A,X,Y
010B  30 01     5             LEAX   1,X        POINT X TO NEXT CHAR
010D  4A        2      CMP2   DECA              ALL CHARS  COMPARE?
010E  27 0C     3             BEQ    CMPOUT     IF SO, IT'S A MATCH, Z=1
0110  E6 A0     6             LDB    ,Y+        GET NEXT BUFFER CHAR.
0112  2B 08     3             BMI    CMPOUT     BRANCH IF BUFFER END, Z=0
0114  E1 80     6             CMPB   ,X+        DOES IT MATCH STRING CHAR?
0116  27 F5     3             BEQ    CMP2       BRANCH IF SO
0118  35 32     9             PULS   A,X,Y      SEARCH FAILED, RESTART SEARCH
011A  20 E4     3             BRA    CMPSTR
011C  35 B2     11     CMPOUT PULS   A,X,Y,PC   FIX STACK, RETURN WITH Z
                       *
         0000                 END
```

Listing 1: 6809 autoincrementing example. This subroutine searches a text buffer for the occurrence of an input string. In autoincrement mode, the value pointed to by the index register is used as the effective address and the index register is then incremented.

to include an additional level of indirection. Even extended addressing can be indirected (as a special indexed addressing mode). Since either stack pointer can be specified as a base address in indexed addressing, the indirect mode allows addresses of data to be passed to a subroutine on a stack as arguments to a subroutine. The subroutine can then reference the data pointed to with one instruction. This increases the efficiency of high level language calls that pass arguments by reference.

M6800 data indicated that quite often the index register was being used in a loop and incremented or decremented each time. This moved the pointer through tables or was used to move data from one area of memory to another (block moves). Therefore, we implemented autoincrement and autodecrement indexed addressing in the M6809. In autoincrement mode the value pointed to by the index register is used as the effective address, and then the index register is incremented. Autodecrement is similar except that the index register is first decremented and then used to obtain the effective address. Listing 1 is an example of a subroutine that searches a text buffer for the occurrence of an input string. It makes heavy use of autoincrementing.

Since the 6809 supports 8 and 16 bit operations, the size of the increment or decrement can be selected by the programmer to be 1 or 2. The post increment, predecrement nature of this addressing mode makes it equivalent in operation to a push and pull from a stack. This allows the X and Y registers to also be used as software stack pointers if the programmer needs more than two stacks. All indexed addressing modes can also contain an extra level of post indirection. Autoincrement and autodecrement are more versatile than the block moves and string commands available on other processors.

Quite often the programmer needs to

calculate the offset used by an indexed instruction during program execution, so we included an index mode that allows the A, B, or D accumulator to be used as an offset to any indexable register. For example, consider fetching a 16 bit value from a two-dimensional array called CAT with dimensions: CAT (100,30). Listing 2 shows the 6809 code to accomplish this fetch. These addressing modes can also be indirected.

Implementation of position independent code was one of the highest priority design goals. The 6800 had limited position independent code capabilities for small programs, but we felt the 6809 must make this type of code so easy to write that most programmers would make all their programs position independent. To do this, an additional long relative (16 bit offset) branch mode was added to all 6800 branches as well as adding program relative addressing. Program relative addressing uses the program counter much as indexing uses one of the indexable registers. This allows all instructions that reference memory to reference data relative to the current program counter (which is inherently position independent). Of course, program relative addressing can be indirected.

The addressing modes of the 6809 have created a processor that has been termed a "programmer's dream machine." To date all the benchmarks we have written for the 6809 are position independent, modular, reentrant and much smaller than comparable programs on other microprocessors. It is easier to write good programs on the 6809 than bad ones!

```
0100                    ORG   $100
0100  108E  1000   4    LDY   #CAT LOAD BASE ADDRESS OF ARRAY
0104  96    32     4    LDA   SUB1 GET FIRST SUBSCRIPT
0106  C6    64     2    LDB   #100 MULTIPLY BY FIRST DIMENSION
0108  3D          11    MUL
0109  D3    33     6    ADDD  SUB2 ADD SECOND SUBSCRIPT
010B  EC    AB     9    LDD   D,Y  FETCH VALUE
```

Listing 2: Array subscript calculations. This 6809 program fetches a 16 bit value from a two-dimensional array called CAT, with dimensions: CAT (100,30).

New or Innovative Instructions

The 6809 does *not* contain dozens of new innovative instructions, and we planned it that way. What we wanted to do was clean up the 6800 instruction set and make it more consistent and versatile. We do not feel a processor with 500 different assembler mnemonics for instructions is better than one with 59 powerful instructions that operate on different data in the same manner. For example, the 6809 contains a transfer instruction of the form TFR R1, R2 that allows transfer of any like-sized registers. There are 42 such valid combinations on the 6809, and clearly one TFR instruction is easier to remember than 42 mnemonics of the form: TAB, TBA, TAP, TXY, etc. Also an exchange instruction (EXG) exists that has identical syntax to the TFR instruction and has 21 valid forms. In the time it took to read three sentences you just learned 63 new 6809 instructions! As another example, we combined the numerous instructions that set and cleared condition code bits on the 6800 into two 6809 instructions that AND or OR immediate data into the condition code register.

Other significant new instructions include the new 16 bit operations. The D register can be loaded, stored, added to, subtracted from, compared, transferred, exchanged, pushed and pulled. All the indexable registers (16 bits) can be loaded, stored and compared. The load effective address instruction can also be used to perform 8 or 16 bit arithmetic on the indexable registers as described later.

Two significant new instructions are the multiple push and multiple pull instructions on the 6809. With one 2 byte instruction, any register or *set* of registers can be pushed

The COMPUTER FACTORY

. . . when software counts 485 Lexington Avenue 750 Third Avenue New York, N.Y. 10017
(212) 687-5001 (212) PET-2001 Foreign order desk - Telex 640055



COMMODORE "PET"

only **$795**

- 14K ROM Operating System
- 8K RAM Memory
- 9" Video Monitor
- Built-In Keyboard
- Digitally controlled tape

commodore — Today's best value in personal computers, along with the latest in peripherals, are always in stock at the COMPUTER FACTORY.



PERIPHERALS FOR PET

- 32K Memory Expansion $595
- 16K EXPANDAPET. $425
- PET to RS232 Serial. $169
- 2 Way Serial/Communication . . $295
- Modem Board for PET. $375
- PET to S100 (4-slot). $160
- Second Cassette Drive. $100
- Parallel Printer Interface $199
- PET Printer (Feb.'79) $695

DUAL DISK (200KB) $1295



apple

APPLE 11
APPLE is a completely contained computer system, with BASIC in ROM, full ASC 11 keyboard in a lightweight molded carrying case. Expandable to 48K
APPLE 11 Systems from $970!

APPLE 11 Disk	$595
APPLESOFT in ROM	$200
Interface Cards	$179

MICROPRODUCTS

- EPROM Prgmr. .... $ 99.95
- EPROM Socket Adpt. .... $ 14.95
- APPLR 11 Modem.......... $129.95

Compucolor II

$1,495 Complete!
16K Model add $300

COMPUCOLOR II Disk-Based Model 3
Advanced hardware and software technology gives you:

- 13 inch Color Display
- Advanced Color Graphics
- 51K Disk Built-In
- 16K ROM Operating System
- 8K RAM User Memory
- 4K RAM Refresh
- 8080A Microcomputer
- RS-232 I/O

Every unit comes with an extended DISK BASIC that has a full file management capability resident in the COMPUCOLOR II in 16K of ROM. Color is fantastic, but COMPUCOLOR II has the power to handle complex tasks and small business applications. An impressive software library supplements your own creativity.

PET MUSIC BOX

Add music and sound effects to your programs. Compose, play, and hear music on your pet. Completely self-contained (no wiring). Free 3 programs including Star Wars theme, sound effects, etc. $39.

FREE $35 of Software with purchase of any computer on this page

SOFTWARE

Word Processing For PET.
With this program in hand, you can create text and add, delete, center, recall lines and move text around on page or between pages. Use either PET or terminal keyboard. $45.00

PET SOFTWARE

BUSINESS FINANCE

Stock Portfolio Analaysis	$ 24.95
Mortgage	15.95
Annual Report Analyzer	22.95
Stock Analyzer	34.95
Bonds	12.95
Checkbook	15.95
Finance	15.95
Data Base (per year) ($30 per monthly tape)	175.00
Stockscreen	29.95
Options	24.95

EDUCATION

Basic Basic	14.95
Diet Planner & Biorhythm	14.95
Basic Math Package	29.95
6502 Assembler in Basic (For PET and Apple II)	24.95
IQ Builder Series 13 to Adult	
IQ Builder-Analogies	9.50
IQ Builder-Vocabulary	12.50
IQ Builder-Number Series	9.50
Preschool IQ Builder	9.50
Step By Step	29.95

GAMES

Osero and Reverse	9.95
Target Pong and Off The Wall	9.95
A Treasure Trove of Games	9.95
Galaxy Games	9.95
Draw Poker	9.95
Black Jack	9.95
Space Fight	9.95
Spacetrek	9.95
Backgammon	14.95
Grand Prix Golf	9.95
Bridge Challenger	14.95
Super Startrek	14.95
Swarm	19.95
Microchess	19.95

RADIO SHACK • PET • SORCERER • APPLE • COMPUCOLOR • ETC.

PRINTERS • PRINTERS • PRINTERS

The COMPUTER FACTORY's extensive inventory and wide selection of computer printers assures you of finding the printer best suited for your needs and specifications. The following printers work well with all known personal computers.

CENTRONICS 779	$1045
AXIOM (Parallel)	$ 395
AXIOM (Serial)	$ 495
EXPANDOR (Tractors)	$ 425
INTEGRAL DATA	$ 795
QUME or DIABLO	$3400

CENTRONICS 700 (Business) $1595

PET TO CENTRONICS PRINTER INTERFACE SWITCH SELECTABLE $199

ANDERSON JACOBSON



ANDERSON JACOBSON
841 I/O Terminal

Ideal for word processing and small businesses.

- ASC 11 Code
- 15 cps Printout
- High Quality Selective Printing
- Use Keyboard for PET
- Reliable heavy duty Mechanism
- Completely Refurbished by A.J.
- Service in 15 Major Cities

Serial - $1195 Parallel - $1095

SORCERER

by EXIDY

- 2-80 Processor
- Keyboard and Numeric Pad
- 8K RAM (up to 32K)
- Serial and Parallel I/O
- Dual Cassette I/O
- 30 x 64 Display
- S-100 Compatible
- 64 User Defined Characters
- 240 x 512 Resolution Graphics

$895

Modular design allows use of both cassettes for programs and ROM PAC cartridges for future language introductions like; APL, COBOL, PILOT, FORTRAN, etc. Add monitor and tape or disk for complete computer system.

SPECIAL
12" Video Monitor for SORCERER ($299 value)
ONLY
$125 with 8K unit
95 with 16K unit
65 with 32K unit

Radio Shack

The folowing program titles are now available for RADIO SHACK TRS 80 Computers.

IQ Builder-Analogies	$ 9.50
IQ Builder-Vocabulary	12.50
IQ Builder-Number Series	9.50
Preschool IQ Builder	9.50
Step By Step	29.95
Microchess	19.95
Annual Report Analyzer	22.50
Stock Analyzer	34.95
Options	24.95
Stockscreen	29.95
Electric Pencil	99.00

BUSINESS COMPUTERS



DATA GENERAL micro NOVA

The ultimate in small business computers when matched with COMPUTER FACTORY's minicomputer; Software - Accounts Receivable/Payable, Inventory Control/Order Entry, General Ledger, Payroll Systems from about **$13,500**

Data General

IMSAI



The low cost solution for all small business problems. A wide variety of software is available for all your needs.
PCS series include dual floppies, 32K RAM, I/O, DOS, BASIC

- PCS-42 (400KB) $2995 • PCS-44 (780KB) $3695

VDP-42 series adds video terminal, keyboard, and VIO to above.
VDP-42 - $4795 • VDP-44 - $4995 • VDP/80 - $6995

APPLE II

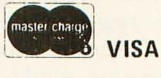
(For Apple II only)

Checkbook	$20.00
Startrek/Starwars	15.00
Color Demo/Breakout	7.50
Applesoft II/Fp Demo	20.00
Hi Res Graphics/Hi Res Shapes	7.50
Ram Test Cassette	7.50
Color Math Demo/Hangman	7.50
Blackjack/Slot Machine	7.50
Biorhythm/Mastermind	7.50
Finance 1-2 Cassette Package	25.00
Datamover/Telepong	7.50

COMPUCOLOR II - $19.95 each disk

MATH TUTOR—with Checkbook, Biorhythms, Math Dice, and Recipe.
CHESS—with Acey Deucey&Line 5
STARTREK—with Lunar Lander,
OTHELLO—with Dice & Concentration
HANGMAN—with Math Tutor, 2 on 10
PERSONAL FINANCE I
PERSONAL FINANCE II
EQUITY
BONDS AND SECURITIES
BLACKJACK

The following are $24.95 each
TEXT EDITOR•ASSEMBLER



VISA

VISA® MC

Pursuant to Rule 206(4)-1(A)(3) of the Securities and Exchange Commission, the scope and usefulness of the above programs are limited, with values in some cases being purely theoretical.

ORDERS ONLY 1-800-223-7318

N.Y. residents add 8% sales tax • Same day shipment on prepaid and credit card orders • Add $10 shipping for computers, $3 for boards, $.25 each cassette tape.

The COMPUTER FACTORY . . . when software counts 485 Lexington Avenue 750 Third Avenue New York, N.Y. 10017
(212) 687-5001 (212) PET-2001 Foreign order desk - Telex 640055
Open Mon-Fri 10-6, Sat 10-4

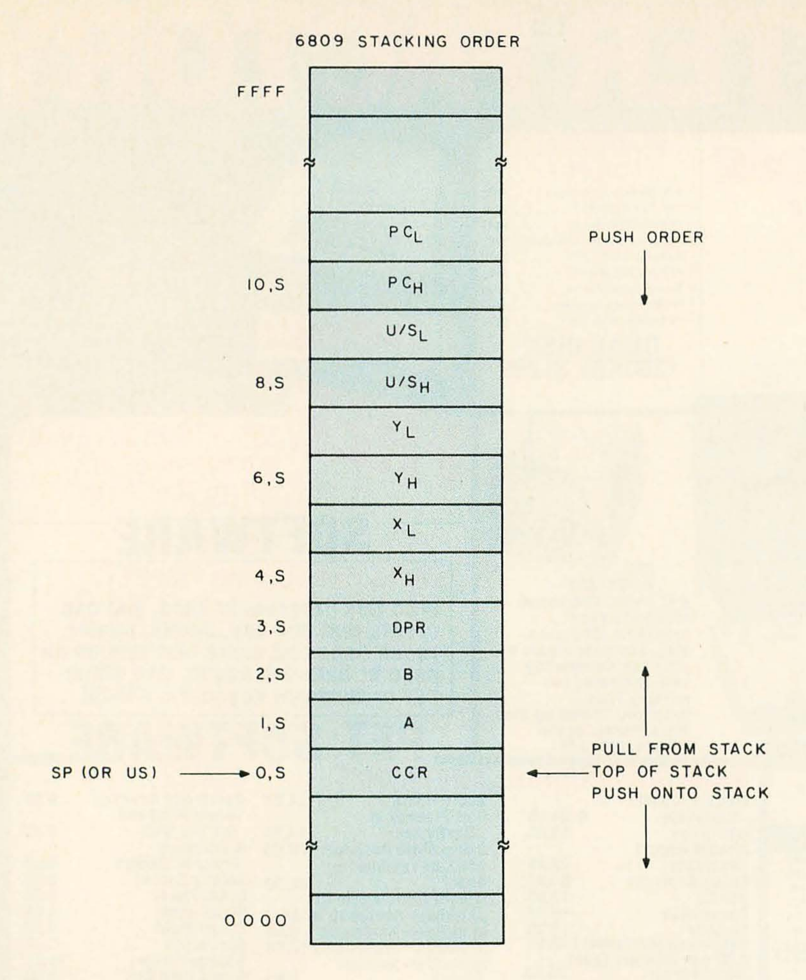


Figure 2: 6809 push/pull and interrupt stacking order.

or pulled from either stack. These instructions greatly decrease the overhead associated with subroutine calls in both assembly and high level language programs. In conjunction with instructions using autoincrement and autodecrement, the 6809 can efficiently emulate a stack computer architecture, which means it should be efficient for Pascal p-code interpreters and the like.

The order in which the registers are pushed or pulled from the stacks is given in figure 2. Note that not all registers need to be pushed or pulled, but that the order is retained if a subset is pushed. This stacking order is also identical to the order used by all hardware and software interrupts.

One new instruction in the 6809 is a sleeper. The load effective address to indexable register (LEA) instruction calculates the effective address from the indexed addressing mode and deposits that address in an indexable register, rather than loading the data pointed to by the effective address, as in a normal load. This instruction was originally created because we wanted a way to let the addressing mode hardware already present in the processor calculate the address of a data object so that it could be passed to a subroutine. After the index addressing modes were completed it was realized the LEA instruction had many more uses, and, once again, allowed us to combine other instructions into one powerful instruction. For example, to add the D accumulator to the Y index register, the instruction is: LEAY D, Y; to add 500 to the U register: LEAU 500, U; and to add 5 to the value in the S register and transfer the sum to the U register: LEAU 5, S.

In writing position independent read only memory programs it is sometimes necessary to reference data in a table within the same read only memory. This is generally a tedious process even in computers that claim to support position independent code because the register that points to the table must eventually contain an absolute address. The LEA instruction, in conjunction with program counter relative addressing, makes this possible with one instruction on the 6809. For example, to put the address of a table DG located in a relative read only memory into indexable register U: LEAU DG, PCR; or, to find out where a position independent read only memory is located: LEAY *, PCR (or TFR PC, Y). Our benchmarks show the LEA to be the most used new 6809 instruction by far.

An unsigned 8 bit by 8 bit to 16 bit multiply was provided for the 6809. The A accumulator contains one argument and the B the other. The result is put back onto the A:B (D) accumulator. A multiply was added because multiplies are used for calculating array subscripts, interpolating values and shifting, as well as for more conventional arithmetic calculations. An unsigned multiply was selected because it can be used to form multiprecision multiplies.

Another facet of good programming practice that we wanted to encourage was the use of operating system calls or software interrupts (SWI). The 6800 SWI has been effectively used by 6800 support software for breakpoints and disk operating system calls. That's nice, but unfortunately there was only one software interrupt, and since Motorola's software used that one, the customer found it difficult to share. The 6809 provides three software interrupts, one of which Motorola promises never to use. It is available for user systems.

One new instruction on the 6809, SYNC, allows external hardware to be synchronized to the software by using one of the interrupt lines. Using this instruction, very tight, fast instruction sequences can be created when it is necessary to process data from very fast input and output devices. Listing 3 gives an example of the use of SYNC. It is assumed that the A side of the peripheral

```
00008 0100                          ORG     $100
00009 0100    B6  F002    5         LDA     PIABC       LOAD PIA CONTROL REG. – SIDE B
00010 0103    84  F7      2         ANDA    #$F7        TURN OFF B-SIDE INTERRUPTS
00011 0105    B7  F002    5         STA     PIABC
00012 0108    8E  3000    3         LDX     #BUFFER     GET POINTER TO BUFFER
00013 010B    C6  80      2         LDB     #128        GET SIZE OF TRANSFER
00014 010D    1A  50      3         ORCC    #$50        DISABLE INTERRUPTS
00015                             * WAIT FOR ANY INTERRUPT LINE TO GO LOW
00016 010F    13          2   LOOP  SYNC                SYNCHRONIZE WITH I/O
00017 0110    B6  F000    5         LDA     PIAAD       LOAD A-SIDE DATA; CLEAR INTERRU
00018 0113    A7  80      6         STA     ,X+         STORE IN BUFFER
00019 0115    5A          2         DECB                DONE?
00020 0116    26  F7      3         BNE     LOOP        BRANCH IF NOT
00021 0118    B6  F002    5         LDA     PIABC       TURN B-SIDE INTERRUPTS BACK ON
00022 011B    8A  08      2         ORA     #$08
00023 011D    B7  F002    5         STA     PIABC
```

Listing 3: Hardware process synchronization using SYNC, a new instruction in the 6809 processor that allows external hardware to be synchronized to the software by using one of the interrupt lines. Very fast instruction sequences can be created using SYNC when it is necessary to process data from very fast input and output devices.

interface adaptor (PIA) is connected to a high speed device that transfers 128 bytes of data to a memory buffer. When the device is ready to send a piece of data, it generates a fast interrupt (FIRQ) from the A side of the peripheral interface adaptor. Program lines 12 and 13 set up the transfer; lines 16 through 20 are the synchronization loop. On each pass through the loop, the program waits at the SYNC instruction until any interrupt line is pulled low. When the interrupt line goes low, the processor executes the next instruction. In order to use SYNC, all other devices tied to any of the interrupt lines must be disabled. For this example it was assumed that the B side of the peripheral interface adaptor also had interrupts enabled; program lines 9 through 11 disable the interrupt and lines 21 through 23 reenable it. Line 14 is included to keep the interrupt by the A side of the peripheral interface adaptor from going to the interrupt routine. Note that interrupts do not need to be enabled for SYNC to work, and in fact are normally disabled.

Another improvement to the instruction set was brought about by inclusion of the hardware signal BUSY. BUSY is high during read/modify/write types of instructions to indicate to shared memory multiprocessors that an indivisable operation is in progress. As shown in figure 3 this fact can be used to turn existing instructions into the LOCK and UNLOCK necessary for mutual exclusion of critical sections of the program, or for allocation of resources.

And lastly, never let it be said the 6809 has no SEX appeal—sign extend, that is. The SEX instruction takes an 8 bit two's complement value in the B accumulator and converts it to a 16 bit two's complement value in the D accumulator by extending the most significant bit (sign bit) of B into A.

Table 4 is a convenient way to look at all the instructions available on the 6809. The notation first page/second page/third page has the following meaning: first page op codes have only one byte of op code. For example: load A immediate has an op code of hexadecimal 86. All second page op codes are preceded by a page op code of hexadecimal 10. For example, the op code for CMPD immediate is hexadecimal 1083 (two bytes). Similarily third page op codes are preceded by a hexadecimal 11. A CMPU immediate is 1183. Some instructions are given two mnemonics as a programmer convenience. For example, ASL and LSL are equivalent. Notice that the long branch op codes LBRA and LBSR were brought onto the first page for increased code efficiency.

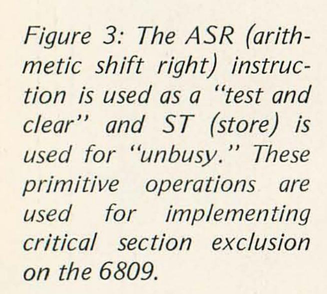
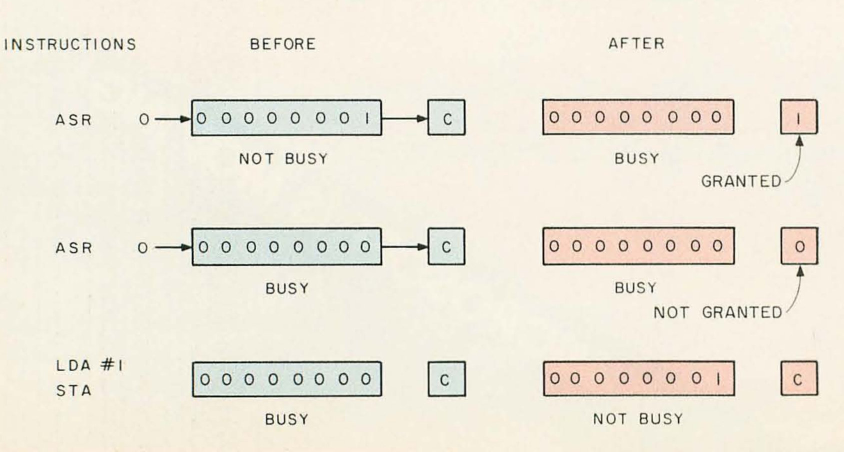


Figure 3: The ASR (arithmetic shift right) instruction is used as a "test and clear" and ST (store) is used for "unbusy." These primitive operations are used for implementing critical section exclusion on the 6809.



Stacks

As mentioned previously, the 6809 has many features that support stack usage. Most modern block structured high level languages make extensive use of stacks. Even though stacks are useful in the typical

textbook example of expression evaluation, their major usage in languages such as Pascal is to implement control structures. Microprocessor users already realize the advantage of a stack in nesting interrupts and subroutine calls. Most high level languages also pass data on the stack and allocate temporary local variables from the stack.

Listing 4 and figure 4 show an example of a high level language subroutine linkage. Before calling the subroutine the caller pushes the addresses of two arguments and the answer on the stack and then executes the jump to subroutine which puts the return program counter on the stack. The subroutine then saves the old stack mark pointer on the stack as well as reserving space on the stack for the local variables for the subroutine. In this example, six locations are used by the subroutine body during calculation. At this point the stack mark pointer is set to a new value for this subroutine. The stack mark pointer is used because the S register may vary during execution of the subroutine body due to local subroutines, etc. It is much more convenient for the compiler to generate offsets to the parameters if the U is used for this purpose instead of the S.

Once U is set it is used to fetch the two arguments using indexed indirect addressing. The subroutine body presumably does something with the arguments and

The complete Motorola 6809 instruction set will be presented in part 2 of this series.

Table 4: 6809 op code map and cycle counts. The numbers by each op code indicate the number of machine cycles required to execute each instruction. When the number contains an I (eg: 4 + I), an additional number of machine cycles equaling I may be required (see table 3). The presence of two numbers, with the second one in parentheses, indicates that the instruction involves a branch. The larger number applies if the branch is taken. The notation first page/second page/third page has the following meaning: first page op codes have only one byte of op code (eg: load A immediate has an op code of hexadecimal 86). All page 2 op codes are preceded by a page op code of hexadecimal 10 (eg: the op code for CMPD immediate is hexadecimal 1083—two bytes). Similarly third page op codes are preceded by a hexadecimal 11. A CMPU immediate is 1183. Some instructions are given two mnemonics as a programmer convenience (eg: ASL and LSL are equivalent). Notice that the long branch op codes LBRA and LBSR were brought onto the first page for increased code efficiency.

Least Significant Four Bits	Most Significant Four Bits																
	DIR		REL		ACCA	ACCB	IND	EXT	IMM	DIR	IND	EXT	IMM	DIR	IND	EXT	
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000 0	6 NEG	PAGE2	3 BRA	4+I LEAX	2	2 NEG	6+I	7	2	4 SUBA	4+I	5	2	4 SUBB	4+I	5	0
0001 1	—	PAGE3	3BRN/ 5LBRN	4+I LEAY	—				2	4 CMPA	4+I	5	2	4 CMPB	4+I	5	1
0010 2	—	2 NOP	3 BHI/ 5(6)LBHI	4+I LEAS	—				2	4 SBCA	4+I	5	2	4 SBCB	4+I	5	2
0011 3	6 COM	2 SYNC	3 BLS/ 5(6)LBLS	4+I LEAU	2	2 COM	6+I	7	4,6,6+I,7 SUBD /	5,7,7+I,8 CMPD /	5,7,7+I,8 CMPU		4	6 ADDD	6+I	7	3
0100 4	6 LSR	—	3 BHS 5(6)(BCC)	5+1/by PSHS	2	2 LSR	6+I	7	2	4 ANDA	4+I	5	2	4 ANDB	4+I	5	4
0101 5	—	—	3 BLO 5(6)(BCS)	5+1/by PULS	—				2	4 BITA	4+I	5	2	4 BITB	4+I	5	5
0110 6	6 ROR	5 LBRA	3 BNE/ 5(6)LBNE	5+1/by PSHU	2	2 ROR	6+I	7	2	4 LDA	4+I	5	2	4 LDB	4+I	5	6
0111 7	6 ASR	9 LBSR	3 BEQ/ 5(6)LBEQ	5+1/by PULU	2	2 ASR	6+I	7	—	4 STA	4+I	5	—	4 STB	4+I	5	7
1000 8	6 ASL (LSL)	—	3 BVC/ 5(6)LBVC	—	2	2 ASL(LSL)	6+I	7	2	4 EORA	4+I	5	2	4 EORB	4+I	5	8
1001 9	6 ROL	2 DAA	3 BVS/ 5(6)LBVS	5 RTS	2	2 ROL	6+I	7	2	4 ADCA	4+I	5	2	4 ADCB	4+I	5	9
1010 A	6 DEC	3 ORCC	3 BPL/ 5(6)LBPL	3 ABX	2	2 DEC	6+I	7	2	4 ORA	4+I	5	2	4 ORB	4+I	5	A
1011 B	—	—	3 BMI/ 5(6)LBMI	6/15 RTI	—				2	4 ADDA	4+I	5	2	4 ADDB	4+I	5	B
1100 C	6 INC	3 ANDCC	3 BGE/ 5(6)LBGE	20 CWAI	2	2 INC	6+I	7	4,6,6+I,7 CMPX /	5,7,7+I,8 CMPY /	5,7,7+I,8 CMPS		3	5 LDD	5+I	6	C
1101 D	6 TST	2 SEX	3 BLT/ 5(6)LBLT	11 MUL	2	2 TST	6+I	7	7 BSR	7 JSR	7+I	8	—	5 STD	5+I	6	D
1110 E	3 JMP	8 EXG	3 BGT/ 5(6)LBGT	—	—		3+I JMP	4	3,5,5+I,6 LDX	/		4,6,6+I,7 LDY	3,5,5+I,6 LDU	/		4,6,6+I,7 LDS	E
1111 F	6 CLR	7 TFR	3 BLE/ 5(6)LBLE	19/20/20 SWI/2/3	2	2 CLR	6+I	7	—	5,5+I,6 STX /	6,6+I,7 STY		—	5,5+I,6 STU /	6,6+I,7 STS		F

FFFE	Restart
FFFC	NMI
FFFA	SWI
FFF8	IRQ
FFF6	FIRQ
FFF4	SWI2
FFF2	SWI3
FFF0	Reserved

Table 5: Hexadecimal addresses of the 6809 restart and interrupt vectors.

finishes with an answer in the D register. The subroutine exit saves this value. It then puts the return address in X and restores the previous stack mark pointer. The whole stack is then cleaned up (deleted) and return is made to the caller.

Motorola 6800 users should note that the stack pointers on the 6809 point to the last value pushed on the stack rather than the next free location, as on the 6800. This was done so that autoincrement and autodecrement would be equivalent to pulls and pushes. For example: STA , -S is equivalent to PSHS A; and LDA , S+ is equivalent to PULS A. This also means the X and Y registers can be used as stack pointers if the programmer desires. For example: STA , -X is a push on a stack defined by X. The possible ambiguity between where the stack pointer points on the 6800 and the 6809 may be less of a problem than it seems, since the 6800's TSX becomes the 6809's TFR S, X without adding 1 and TXS becomes a TFR X,S without subtracting 1 — think about it. The only danger is in programs that used the stack pointer as an index register. In these programs the stack pointer may point one location away from where it did previously.

```
00006 0500 34  40       6  SUBR PSHS U        SAVE OLD STACK MARKER
00007 0502 32  66       5       LEAS 6,S      RESERVE LOCAL STORAGE
00008 0504 1F  43       6       TFR  S,U      GET NEW STACK MARKER
00009 0506 EC  D8  0E 10        LDD  [14,U]   GET ARGUMENT 1
00010 0509 AE  D8  0C 10        LDX  [12,U]   GET ARGUMENT 2
00011                      *
00012                      *  SUBROUTINE BODY
00013                      *
00014 050C ED  D8  0A 10        STD  [10,U]   SAVE ANSWER
00015 050F AE  48       6       LDX  8,U      GET RETURN ADDRESS
00016 0511 EE  46       6       LDU  6,U      RESTORE U'
00017 0513 32  E8  10  6        LEAS 16,S     POP EVERYTHING OFF STACK
00018 0516 6E  84       3       JMP  ,X       RETURN
```

Listing 4: Use of stacks on the 6809 processor. In this typical high level language subroutine example, U' and S' are the mark stack pointer and hardware stack pointer, respectively, just prior to the call. U and S are the same registers during execution of the subroutine body. Before calling the subroutine the caller pushes the addresses of two arguments and the answer on the stack and then executes the jump to subroutine which puts the return program counter on the stack. The subroutine then saves the old stack mark pointer on the stack as well as reserving space on the stack for the local variables for the subroutine (see figure 4).

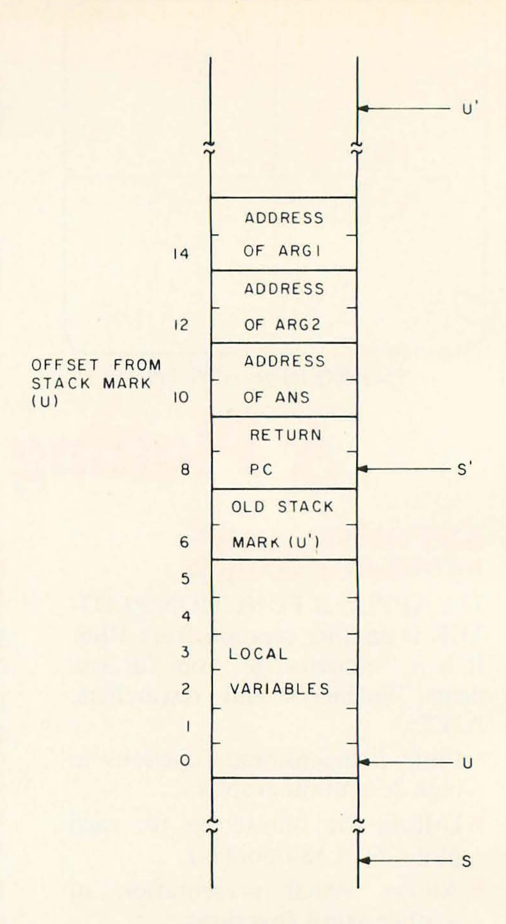


Figure 4: Illustration of the high level language subroutine example in listing 1.

Interrupts

The 6809 has three fully vectored hardware interrupts. The nonmaskable interrupt (NMI) and maskable interrupt (IRQ) are the same as the 6800's NMI and IRQ. The new interrupt is the fast maskable interrupt, or FIRQ, that stacks the program counter and condition code register only on interrupt. Table 5 gives the addresses of the interrupt vectors for the 6809.

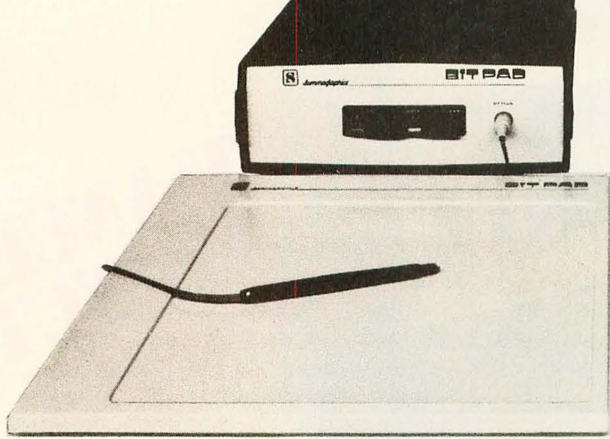
A new signal (IACK) has been added that is available anytime an interrupt vector is fetched. This signal together with address bus lines A1 through A3 can be used to implement an interrupt scheme in which each device supplies its own interrupt vector.

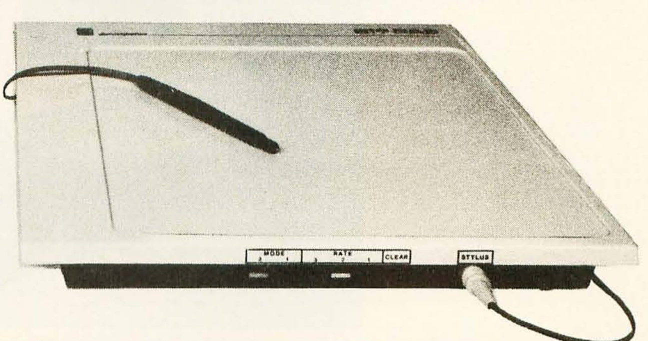
The interrupt control and prioritization logic of the 6809 have been defined very carefully — no redundant or indeterminate conditions can exist when several interrupts occur simultaneously. The details of this interrupt structure are precisely defined in Motorola documentation for the 6809.

Part 2, entitled "Instruction Set Dead-Ends, Old Trails and Apologies," will be a question and answer discussion about the design philosophy that went into the 6809.■

Our Two Bits

Bit Pad Bit Pad One
Bit Pad Bit Pad One
Bit Pad Bit Pad One
Bit Pad Bit Pad One
Bit Pad Bit Pad One
Bit Pad One
Bit Pad One





Bit Pad™ is the low-cost digitizer for small computer systems. Better than a joystick or keyboard for entering graphic information, it converts any point on a page, any distance into its digital equivalents. It's also a menu for data entry. You assign a value or an instruction to any location on the pad. At the touch of a stylus, it's entered into your system.

Who can use it? Anyone from the educator and the engineer to the hobbyist and the computer games enthusiast. The data structure is byte oriented for easy compatibility with small computers, so you can add a power supply, stand alone display, cross-hair cursor and many other options.

Bit Pad by Summagraphics. The leading manufacturer of data tablet digitizers. Bit Pad. The only words you need to say when considering digitizers.

$1,000 creativity prize. Just write an article on an original Bit Pad application and submit it to any national small-computer periodical. If the editors publish it—and the decision is solely theirs—Summagraphics will pay you $1,000.



35 Brentwood Ave., Box 781, Fairfield, CT 06430
Phone (203) 384-1344. TELEX 96-4348

Dealer inquiries invited

Circle 356 on inquiry card.

An advanced desktop data system for $1,995? Quitcherkiddin,* TANO.

We're not kiddin'.
Outpost 11 is exactly that. You get:

- A ruggedly designed unit, intended for heavy use
- Full ASCII keyboard
- 24 x 80 character CRT
- 32K bytes RAM
- M6800 CPU
- Mini-floppy disk drive and controller
- BASIC software package
- Over 20 applications packages available
- User documentation

Just like this, for only



$1995 COMPLETE

With Outpost 11 and the BASIC software package, you can:
Do full floating point BCD math ● Program SAVE and LOAD with file names ● Program MERGE and CHAIN ● ON ERROR statement ● String variables and functions ● Trig and math functions ● GET, PUT and FIELD statements ● Floppy file processor.

You've been waiting a long time for a Data System with all these features at a price like this. Now it's yours. No kiddin'.

*Quit your kidding

Order Form

Quitcherkiddin', TANO. I'm calling your hand. Send me ____________ Outpost 11's as described in your ad. Here's my deposit of $200 each, plus $35 freight and insurance. I'll pay the balance on delivery.

Name ____________

Company/Title ____________

Shipping Address ____________

Telephone ____________ Signature ____________

Charge my: ☐ American Express; ☐ Visa; ☐ Master Charge: #____________

Mail to:

TANO Corporation
4521 W. Napoleon Avenue
Metairie, La. 70001
(504) 888-4884
TWX 810-591-5229

Sales Offices: Los Angeles, CA—213/426-7375 • Montreal, Canada—514/934-0000 • Boston, MA—617/969-4650 • Dallas, TX—214/358-1307 • Houston, TX—713/461-0038 • Denver, Col.—303/841-2788 • San Francisco, CA—408/377-7001 • Chicago, IL—312/830-0060 • Atlanta, GA—404/252-6609 • Washington, DC—301/589-2802 •

Circle 358 on inquiry card.



"BOOKS OF INTEREST TO COMPUTER PEOPLE"

More BYTE BOOKS in your future...

...And the future

THE BYTE BOOK OF COMPUTER MUSIC combines the best computer music articles from past issues of BYTE Magazine with exciting new material—all written for the computer experimenter interested in this fascinating field.

You will enjoy Hal Chamberlin's "A Sampling of Techniques for Computer Performance of Music", which shows how you can create four-part melodies on your computer. For the budget minded, "A $19 Music Interface" contains practical tutorial information—and organ fans will enjoy reading "Electronic Organ Chips For Use in Computer Music Synthesis".

New material includes "Polyphony Made Easy" and "A Terrain Reader". The first describes a handy circuit that allows you to enter more than one note at a time into your computer from a musical keyboard. The "Terrain Reader" is a remarkable program that creates random music based on land terrain maps.

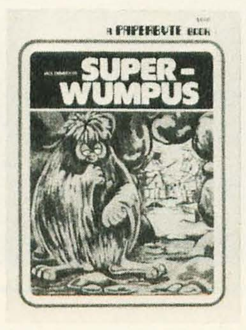
Other articles range from flights of fancy about the reproductive systems of pianos to Fast Fourier transform programs written in BASIC and 6800 machine language, multi-computer music systems, Walsh Functions, and much more.

For the first time, material difficult to obtain has been collected into one convenient, easy to read book. An ardent do-it-yourselfer or armchair musicologist will find this book to be a useful addition to the library.



ISBN 0-931718-11-2
Editor: Christopher P. Morgan
Pages: approx. 128
Price: **$10.00**

SUPERWUMPUS is an exciting computer game incorporating the original structure of the WUMPUS game along with added features to make it even more fascinating. The original game was described in the book **What To Do After You Hit Return,** published by the People's Computer Company. Programmed in both 6800 assembly language and BASIC, SUPERWUMPUS is not only addictively fun, but also provides a splendid tutorial on setting up unusual data structures (the tunnel and cave system of SUPERWUMPUS forms a dodecahedron). This is a **PAPERBYTE™** book.



ISBN 0-931718-03-1
Author: Jack Emmerichs
Pages: 56
Price: **$6.00**

TINY ASSEMBLER 6800, Version 3.1 is an enhancement of Jack Emmerichs' successful Tiny Assembler. The original version (3.0) was described first in the April and May 1977 issues of BYTE magazine, and later in the **PAPERBYTE™** book **TINY ASSEMBLER 6800 Version 3.0.**



In September 1977, BYTE magazine published an article entitled, "Expanding The Tiny Assembler". This provided a detailed description of the enhancements incorporated into Version 3.1, such as the addition of a "begin" statement, a "virtual symbol table", and a larger subset of the Motorola 6800 assembly language.

All the above articles, plus an updated version of the user's guide, the source, object and PAPERBYTE™ bar code formats of both Version 3.0 and 3.1 make this book the most complete documentation possible for Jack Emmerichs' Tiny Assembler.

ISBN 0-931718-08-2
Author: Jack Emmerichs
Pages: 80
Price: **$9.00**

A walk through this book brings you into **Ciarcia's Circuit Cellar** for a detailed look at the marvelous projects which let you do useful things with your microcomputer. A collection of more than a year's worth of the popular series in BYTE magazine, **Ciarcia's Circuit Cellar** includes the six winners of BYTE's On-going Monitor Box (BOMB) award, voted by the readers themselves as the best articles of the month: **Control the World** (September 1977), **Memory Mapped IO** (November 1977), **Program Your Next EROM in BASIC** (March 1978), **Tune In and Turn On** (April 1978), **Talk To Me** (June 1978), and **Let Your Fingers Do the Talking** (August 1978).

Each article is a complete tutorial giving all the details needed to construct each project. Using amusing anecdotes to introduce the articles and an easy-going style, Steve presents each project so that even a neophyte need not be afraid to try it.



ISBN 0-931718-07-4
Author: Steve Ciarcia
Pages: approx. 128
Price: **$8.00**

is right now!

BASEX, a new compact, compiled language for microcomputers, has many of the best features of BASIC and the 8080 assembly language—and it can be run on any of the 8080 style microprocessors: 8080, Z-80, or 8085. This is a **PAPERBYTE™** book.

Subroutines in the **BASEX** operating system typically execute programs up to five times faster than equivalent programs in a BASIC interpreter—while requiring about half the memory space. In addition, **BASEX** has most of the powerful features of good BASIC interpreters including array variables, text strings, arithmetic operations on signed 16 bit integers, and versatile IO communication functions. And since the two languages, BASEX and BASIC, are so similar, it is possible to easily translate programs using integer arithmetic data from BASIC into BASEX.

The author, Paul Warme, has also included a BASEX Loader program which is capable of relocating programs anywhere in memory.



ISBN 0-931718-05-8
Author: Paul Warme
Pages: 88
Price: **$8.00**

PROGRAMMING TECHNIQUES is a series of **BYTE BOOKS** concerned with the art and science of computer programming. It is a collection of the best articles from BYTE magazine and new material collected just for this series. Each volume of the series provides the personal computer user with background information to write and maintain programs effectively.



The first volume in the Programming Techniques series is entitled **PROGRAM DESIGN.** It discusses in detail the theory of program design. The purpose of the book is to provide the personal computer user with the techniques needed to design efficient, effective, maintainable programs. Included is information concerning structured program design, modular programming techniques, program logic design, and examples of some of the more common traps the casual as well as the experienced programmer may fall into. In addition, details on various aspects of the actual program functions, such as hashed tables and binary tree processing, are included.

ISBN 0-931718-12-0
Editor: Blaise W. Liffick
Pages: 96
Price: **$6.00**

SIMULATION is the second volume in the Programming Techniques series. The chapters deal with various aspects of specific types of simulation. Both theoretical and practical applications are included. Particularly stressed is simulation of motion, including wave motion and flying objects. The realm of artificial intelligence is explored, along with simulating robot motion with the microcomputer. Finally, tips on how to simulate electronic circuits on the computer are detailed.

ISBN 0-931718-13-9
Editor: Blaise W. Liffick
Pages: approx. 80
Price: **$6.00**
Publication: Winter 1979

RA6800ML: AN M6800 RELOCATABLE MACRO ASSEMBLER is a two pass assembler for the Motorola 6800 microprocessor. It is designed to run on a minimum system of 16 K bytes of memory, a system console (such as a Teletype terminal), a system monitor (such as Motorola MIKBUG read only memory program or the ICOM Floppy Disk Operating System), and some form of mass file storage (dual cassette recorders or a floppy disk).

The Assembler can produce a program listing, a sorted Symbol Table listing and relocatable object code. The object code is loaded and linked with other assembled modules using the **Linking Loader LINK68.** (Refer to **PAPERBYTE™** publication **LINK68: AN M6800 LINKING LOADER** for details.)

There is a complete description of the 6800 Assembly language and its components, including outlines of the instruction and address formats, pseudo instructions and macro facilities. Each major routine of the Assembler is described in detail, complete with flow charts and a cross reference showing all calling and called-by routines, pointers, flags, and temporary variables.

In addition, details on interfacing and using the Assembler, error messages generated by the Assembler, the Assembler and sample IO driver source code listings, and **PAPERBYTE™** bar code representation of the Assembler's relocatable object file are all included.

This book provides the necessary background for coding programs in the 6800 assembly language, and for understanding the innermost operations of the Assembler.

ISBN 0-931718-10-4
Author: Jack E. Hemenway
Pages: 184
Price: **$25.00**

to order books see next page .....

LINK68: AN M6800 LINKING LOADER is a one pass linking loader which allows separately translated relocatable object modules to be loaded and linked together to form a single executable load module, and to relocate modules in memory. It produces a load map and a load module in Motorola MIKBUG loader format. The Linking Loader requires 2 K bytes of memory, a system console (such as a Teletype terminal), a system monitor (for instance, Motorola MIKBUG read only memory program or the ICOM Floppy Disk Operating System), and some form of mass file storage (dual cassette recorders or a floppy disk).

It was the express purpose of the authors of this book to provide everything necessary for the user to easily learn about the system. In addition to the source code and **PAPERBYTE™** bar code listings, there is a detailed description of the major routines of the Linking Loader, including flow charts. While implementing the system, the user has an opportunity to learn about the nature of linking loader design as well as simply acquiring a useful software tool.

ISBN 0-931718-09-0
Authors: Robert D. Grappel
& Jack E. Hemenway
Pages: 72
Price: **$8.00**
Winter 1979

TRACER: A 6800 DEBUGGING PROGRAM is for the programmer looking for good debugging software. TRACER features single step execution using dynamic break points, register examination and modification, and memory examination and modification. This book includes a reprint of "Jack and the Machine Debug" (from the December 1977 issue of BYTE magazine), **TRACER** program notes, complete assembly and source listing in 6800 assembly language, object program listing, and machine readable **PAPERBYTE™** bar codes of the object code.

ISBN 0-931718-02-3
Authors: Robert D. Grappel
& Jack E. Hemenway
Pages: 24
Price: **$6.00**

MONDEB: AN ADVANCED M6800 MONITOR-DEBUGGER has all the general features of Motorola's MIKBUG monitor as well as numerous other capabilities. Ease of use was a prime design consideration. The other goal was to achieve minimum memory requirements while retaining maximum versatility. The result is an extremely versatile program. The size of the entire MONDEB is less than 3 K.

Some of the command capabilities of MONDEB include displaying and setting the contents of registers, setting interrupts for debugging, testing a programmable memory range for bad memory locations, changing the display and input base of numbers, displaying the contents of memory, searching for a specified string, copying a range of bytes from one location in memory to another, and defining the location to which control will transfer upon receipt of an interrupt. This is a **PAPERBYTE™** book.

ISBN 0-931718-06-6
Author: Don Peters
Pages: 88
Price: **$5.00**

BAR CODE LOADER. The purpose of this pamphlet is to present the decoding algorithm which was designed by Ken Budnick of Micro-Scan Associates at the request of BYTE Publications, Inc., for the **PAPERBYTE™** bar code representation of executable code. The text of this pamphlet was written by Ken, and contains the general algorithm description in flow chart form plus detailed assemblies of program code for 6800, 6502 and 8080 processors. Individuals with computers based on these processors can use the software directly. Individuals with other processors can use the provided functional specifications and detail examples to create equivalent programs.

ISBN 0-931718-01-5
Author: Ken Budnick
Pages: 32
Price: **$2.00**

BYTE BOOKS Division • 70 Main Street • Peterborough, New Hampshire 03458

Name ____________ Title ____________ Company ____________

Street ____________ City ____________ State/Province ____________ Code ____________

☐ Check enclosed in the amount of $ ________
☐ Bill Visa ☐ Bill Master Charge Card number ________________ Expiration Date ______

Please send the books I have checked.

☐ **Computer Music $10.00**
☐ **SUPERWUMPUS $6.00**
☐ **Tiny Assembler (3.1) $9.00**
☐ **Circuit Cellar $8.00**
☐ **BASEX $8.00**
☐ **Program Design $6.00**

☐ **Simulation $6.00**
☐ **RA6800ML $25.00**
☐ **Link68 $8.00**
☐ **TRACER $6.00**
☐ **Mondeb $5.00**
☐ **Bar Code Loader $2.00**

Add **50¢** per book to cover postage and handling

Please allow 6–8 weeks for processing your order.

Book Reviews

Financial Analysis and Business Decisions on the Pocket Calculator
by Jon M Smith
John Wiley and Sons, New York 1976
313 pages
$12.95

For many people, mathematics applied to their financial affairs consists of reconciling checking account statements, filling out income tax forms, trying to pay all their bills and live within their means, scheming to have more money, and once in a lifetime retirement planning, but not much more.

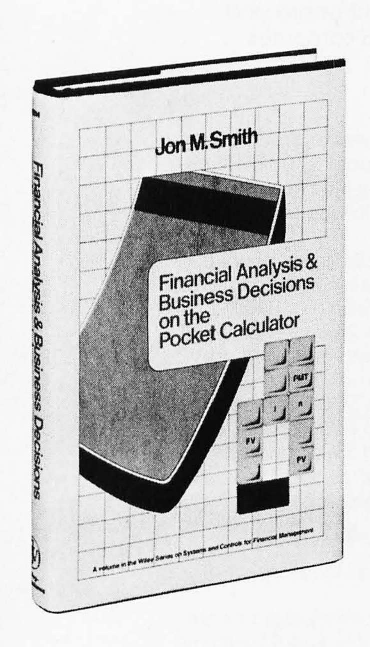
When I unexpectedly received a review copy of Jon Smith's second book, I thought it would be dull. It looked like a particularly thorough version of eighth grade arithmetic, but not difficult enough to excuse me from reading all of it. Fortunately, this first impression was wrong. Smith has managed to make the subject quite interesting, if not actually entertaining. Once into the book, it became progressively harder to put aside, and I was rather sorry to come to its end.

This is essentially a guidebook to the mathematical aspects of scheming to make more money (by lawful means), emphasizing how any pocket calculator (but particularly the "financial" kind) can be of assistance.

The chapters concerning business decisions provide a quick and fascinating introduction to decision trees, certain applications of probability and statistics that lead to systems analysis, and forecasting techniques including linear regression. Yes, some of the financial calculators can do linear regression with very few keystrokes.

One third of the book is devoted to casebook examples giving complete keystroke sequences for a variety of calculators. Where appropriate, general sequences are given for 4 function calculators of both algebraic and reverse Polish notation (RPN) types. Problems involving compound interest tend to go beyond their capability for easy solution, and are often only shown solved by financial models, specifically the HP-22, 70 and 80, Novus 6020 and Rockwell 204.

Despite the strong capabilities of their built-in (or hardware preprogrammed) functions, these machines are given quite a run for their money with some rather long programs (keystroke sequences). Of course, the more expensive calculators with the most such functions generally have the easiest time of it (although not always), need the fewest steps, and have the least need for trial approximations.



The text emphasizes understanding of the necessary computations so that, if your calculator is not one of those above, a good understanding of its instructions should enable you to modify the keystroke sequences as I did when needed. For simple calculators, Smith does not even assume you have a percent (%) key, so it is often possible to shorten the sequence. A brief section covers general principles for using programmable calculators.

Much is made of the book's interdisciplinary aspects in which Smith shows how the same problem is approached through different terminology by specialists in different fields. He simplifies all the financial analysis to five parameters: present value, future value, number of compounding periods, interest rate per compounding period, and amount of equal periodic payments. These are combined into only 12 different equations: three each for investment yield comparisons (solving for interest rate) and investment horizon comparisons (solving for number of compounding periods); and two each for net present value analysis (solving for present value), growth comparison analysis (solving for future value) and affordability analysis (solving for amount of periodic payment). Not all financial calculators have all twelve equations preprogrammed.

In the special chapters on business statistics and business systems analysis, Smith really seems to hit his stride, and these may leave you breathless. In addition, there are four special but more easygoing chapters on consumer finance, merchandising calculations, real estate calculations and programmable calculators. The four appendices are: the "HUD Guide to Real Estate Settlement Costs," tables of binomial probabilities and Gaussian distributions, and some text on concepts of time and money (borrowed from Hewlett-Packard).

The chapter on programmable calculators includes the only illustration of solving continuous compounding, although this is mentioned in two other places. The illustration uses differential equations. Surprisingly, Smith does not show the simple formula: $FV = PV\ (e^{n \times i})$, where n is in years and i is the annual interest rate (not the usual rate per compounding period) as a decimal, and e is the base for the natural logarithms.

There is also little treatment of calculating errors. For instance, taking the square root of e for a continuous compounding problem, eight significant figures are needed in the answer to be sure of rounding to the nearest penny. Of two 8 place display calculators I have at hand, one shows the square root of e to seven places, the other to eight. If logarithm and antilogarithm conversions are made (compared to straight power, root, multiplication and division processes such as in the power series when either method could be used), one or two places of accuracy may be lost, depending on the specific example and the calculator used.

Perhaps deliberately, the author does not show how to compute a year's daily compounding on the ordinary 4 function calculator, which is easy on a 360 day basis and difficult for 365 days (see "Just Look at What a Little Calculator Can Do!", *Changing Times,* January 1977, pages 17 and 18). This involves raising the daily interest multiplier to the 360th power, a remarkable feat accomplished with fair accuracy on an 8 place calculator, as long as the annual interest rate is not too low and not too many places of accuracy are needed in the answer. It is also possible to correct for one or two leap years.

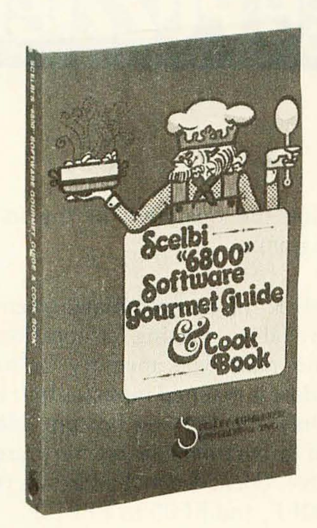
Computers can pinch pennies more easily than can financial calculators. On daily compounding, for example, you may never receive any fractional cents from day to day, or you may be given a full cent for half or more and lose each less than half, or rounding may be done only at the end. I believe this is why different banks can offer certificates at the same annual rate, both compounded daily, but with different effective annual yields. A few calculators round up or down, called rounding or rounding off, but most of them truncate the results. Truncation tends to produce somewhat larger calculating errors than rounding. Some desktop calculators have a switch to allow choosing either when using a fixed number of decimal places.

At least one scientific calculator includes preprogrammed financial functions. Some remarkably inexpensive financial calculators have come on the market. Programmable calculators can be set to do financial equations, but often require many programming steps. At least two manufacturers have combined the power of preprogrammed financial functions with the versatility of programmable keystroke sequences (the Novus 6025, Texas Instruments MBA), which should provide increased convenience and reduced likelihood of operator error in repetitive calculations.

Unfortunately, some errors have crept into the book. If you have the first printing, send a self-addressed stamped envelope to Gerald R Galbo, editor, John Wiley and Sons Inc, 605 Third Av, New York NY 10016, for a listing of the corrections and the author's meticulous changes.

John F Sprague
143 Myrtle Av
Allendale NJ 07401■

Scelbi "6800" Software Gourmet Guide & Cook Book
by Robert Findley
Scelbi Computer Consulting Inc
Milford CT, 1976
$9.95



This book is the first of a series of text and program collections for the Motorola 6800 produced by Scelbi Computer Consulting Inc, authors of a popular series of similar books for the Intel 8080. The book can be viewed either as an educational aid for 6800 users or as a source of extensively documented programs: the ratio of tutorial or narrative text to program listings and flowcharts is about 3 to 1.

The first two chapters deal with the 6800 instruction set and basic 6800 programming techniques, such as the use of page zero, relocatability of programs, breakpoints, and use of the stack. Multiple precision arithmetic is then covered in some depth, along with time delay routines and random number generators. A full chapter is devoted to an extensive discussion of floating point routines, and another chapter to long-precision decimal arithmetic routines; both are complete with source code listings. The coverage of input/output and interrupt processing includes techniques for a "software UART" and a brief discussion of interrupt priorities. The last chapter presents sorting and table searching routines. Finally, six appendices provide useful reference information, including a relocatable object code listing of the floating point routines.

This book would be a useful reference source for anyone who is developing 6800 applications software and is highly recommended to the user who is learning assembly language programming on the 6800.

Dan Fylstra
22 Weitz St #3
Boston MA 02134■

Machine Language Puzzler

Memory Meanderings

Christopher Strangio
CAMI Research
43 Bailey Rd
Watertown MA 02172

Address	Hexadecimal Code	Label	Operator	Operand
FFFA	31 FD FF		LXI	SP, FFFD
FFFD	CD FD FF	LOOP	CALL	LOOP

Describe the operation of the following 8080 program if:

(a) 64 K of programmable memory is available, starting at location 0000 (no read only memory or memory address mapped input/output is used.)
(b) Only the top and bottom 256 bytes of programmable memory are available. (Hexadecimal addresses 0000 to 00FF, and FF00 to FFFF.)

Turn to page 181 for the answers, but not until you've figured out what you think they are. For reference materials see the Intel 8080 specifications or any of a number of books on that processor.■

BYTE's Bugs

Functional Bug

We received the following from F R Ruckdeschel:

The author apologizes for making an obvious error in the coefficients shown on table 1 of "Functional Approximations" (November 1978 BYTE, page 38) for the expansion of sin(x). The mistake occurred in transforming the coefficients for the optimal series representation of $\sin(\pi x/2)$ to the more useful sin(x) form. The transformation was so simple that nothing (?) could go wrong. The correct coefficients, to the digit shown, are given below. The accuracy of the approximation is better than 2×10^{-7} over the range indicated.

Term	MacLaurin Coefficients	Optimal Coefficients
x	1.00000000	1.0000000
x^3	−0.16666667	−0.1666666
x^5	0.0083333333	0.008333026
x^7	−0.00019841270	−0.0001980742
x^9	0.000002755732	0.000002601887
x^{11}	$-2.5052109 \times 10^{-8}$	
x^{13}	$1.6059045 \times 10^{-10}$	

■

Motor Bug

A small bug stepped into the stepper motor described in "I've Got You In My Scanner! A Computer Controlled Stepper Motor Light Scanner" (November 1978 BYTE, page 86). The part number of the motor should be K82701-P2, not K82944-M1, as stated. For price and ordering information, contact North American Philips, Sales Dept, Cheshire CT 06401, (203) 272-0301.■

Out of Phase

In Leonard H Anderson's article "Linear Circuit Analysis" in October 1978 BYTE, the table on page 102, "Complex Number Arithmetic," contains two mathematical errors:

1. The correct formula for magnitude should be MAGN = $\sqrt{(A^2 + B^2)}$
2. The correct formula for phase angle should be PHA = ARCTAN (B/A).

Attempting to use the formulas given in the article will result in chaos.

D M Graham
2149 Scarboro Av
Vancouver BC
CANADA V5P 2L2

Oops! Mr Graham has detected an obvious error. We hope our college mathematics teachers weren't looking
. . . .CM■

Technical Forum

Comments on S-100 Bus Extension

John Walker
Marinchip Systems
16 St Jude Rd
Mill Valley CA 94941

The letter from Olav Naess (Technical Forum, August 1978 BYTE, page 12) suggesting the use of the data in and data out lines of the S-100 bus as a bidirectional data bus was especially interesting in light of the announcement of a commercial product that operates this way, and adoption of this mode of 16 bit operation in the proposed standard for the S-100 bus.

In August 1977, I was involved in designing the prototype of a processor board that interfaced the Texas Instruments TMS9900 16 bit processor to the S-100 bus. Since the design was to work with most existing S-100 boards, it included logic to multiplex the 16 bit data transfers into pairs of conventional 8 bit memory cycles. Since performing this multiplexing imposed a substantial performance penalty on the processor compared to operation with 16 bit parallel memories, I wanted to allow the processor to also work with 16 bit wide memories, ideally intermixed with normal S-100 memories.

After discarding several schemes such as having a separate 16 bit wide bus along the top of the card, and splitting the bus into two separate mother boards, I hit on the idea of using the data in and data out buses as a bidirectional bus. The final design was as follows.

At the start of a memory cycle, the processor asserts a line, informing the memory that the processor is capable of 16 bit transfers. If the memory addressed is also capable of 16 bit operation, it responds with a signal informing the processor of this fact. When the processor sees the 16 bit mode reply from the memory, it initiates a parallel 16 bit transfer instead of the normal pair of 8 bit transfers. Since a conventional 8 bit S-100 memory is not connected to the 16 bit reply line, and the processor pulls this line to the 8 bit value, normal 8 bit memories will run with the processor with no modifications in multiplexed mode.

The processor's 16 bit request line allows the design of memories that operate in either 8 or 16 bit mode. Such design is desirable because memories so designed can be used both with regular 8 bit processors or with the new 16 bit processors. Also, such a memory is required in order to use an 8 bit direct memory access device in a system with a 16 bit processor and memory.

The Marinchip Systems M9900 processor was designed to run in the manner described above. A prototype of the processor with 16 bit mode operation was tested in January 1978, and the production model of the processor was demonstrated at the West Coast Computer Faire in March 1978, running with mixed 8 and 16 bit memories and an 8 bit direct memory access disk controller. At the Faire, our scheme for 16 bit operation was presented at the S-100 standards meeting. The S-100 standards committee later adopted it in the proposed standard.

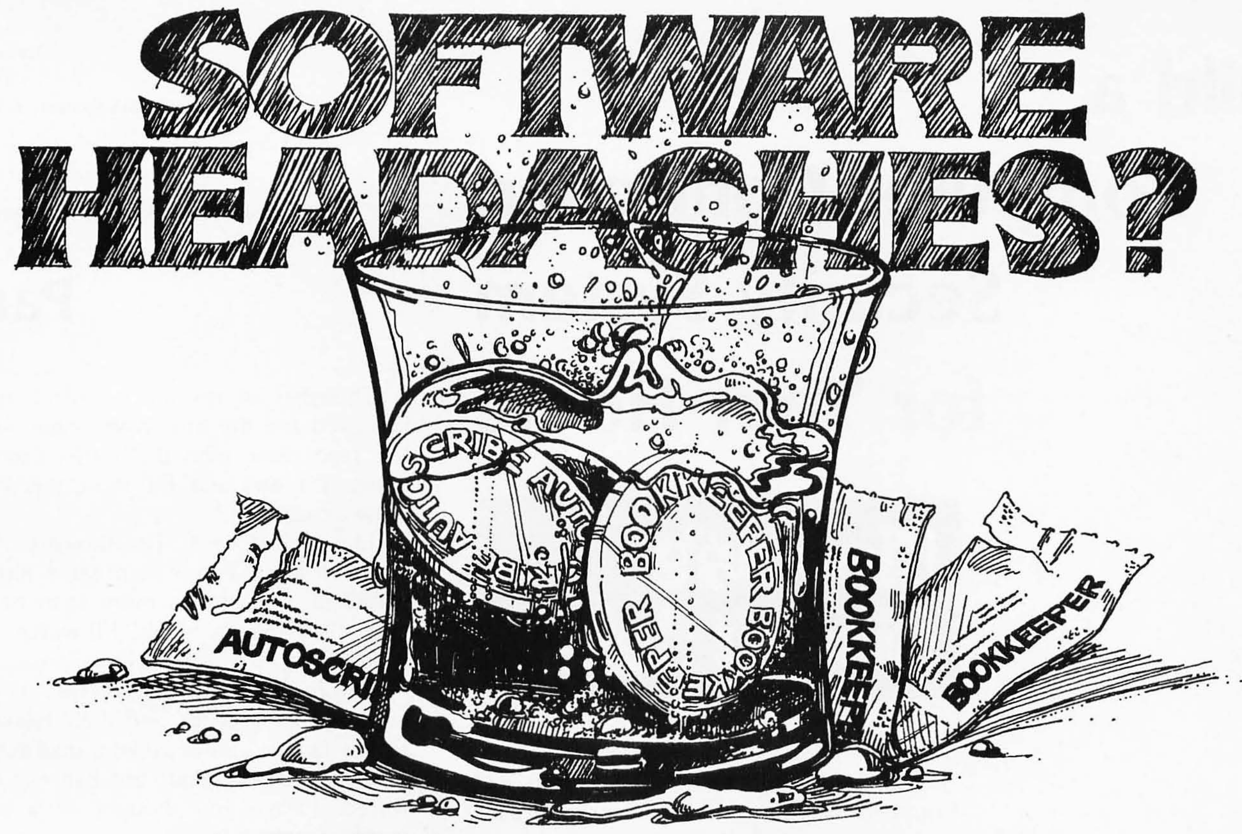
The bidirectional 16 bit bus proposal, as integrated into the proposed standard, has many advantages. It requires no new hardware signal routing (compared to a separate 16 bit bus, or a split mother board). It requires only two new bus lines: the 16 bit request from the processor, and the 16 bit acknowledgement from the memory. Finally, and most importantly, it allows a smooth transition from a primarily 8 bit world to the coming 16 bit era. Users' investment in 8 bit memory, input/output, and direct memory access devices is protected, as these devices continue to run without modification, while 16 bit processors and devices can be added as they become available. The 16 bit processor, when executing out of 16 bit memory, operates at its full rated speed.

To answer the objections raised to this proposal by John C McCallum in the same issue, allowing 8 bit memory boards to operate in their original mode allows them to work with a 16 bit processor with no problems. If a memory board works with an 8080, it will work in 8 bit mode with the new processor. Processor Technology's parallel bus affects only the SOL computer, as far as I know, and not any of their accessory boards. In any case, their 3P+S, VDM-1, and 8KRA memory work fine with the M9900 processor. Since the SOL has the processor integrated in the system rather than on a separate card, users are not likely to change it anyway. Finally, nobody expects anybody to rewire anything under the 16 bit proposal. The most attractive facet of the proposal is that it allows existing hardware to run as it does now, while allowing new hardware to run with unimpaired performance.

If the S-100 bus is to remain the dominant bus for modular system construction, it must accommodate the existing and forthcoming 16 bit processors. I feel the proposed 16 bit standard is the best route to this goal.■

REFERENCE

Morrow, G, and Fullmer, H, "Proposed Standard for the S-100 Bus," *Computer,* May 1978, page 84.



HERE'S WHAT DEALERS RECOMMEND FOR RELIEF....

Russell F. Roth, President, Roth Computer Processing, Indianapolis, IN: "**Bookkeeper** and **Autoscribe** are flexible enough for almost any business application. MicroSource narrows the applications gap between micros and the other guys . . . refreshing for packaged systems."

Chuck Faso, Owner, Computerland of Niles, IL: "You cannot find more for your money anywhere. **Autoscribe** is the easiest to operate I've seen. It does everything you would want to do with word processing. There's just no competition at this price."

Norm Dinnsen & Dave McDonough, Owners, Computerland of San Diego: "We like to sell systems we can depend on. That's why we sell MicroSource. MicroSource software is more business-oriented, the systems are a lot cleaner and they use good dependable equipment.

"MicroSource has the best documentation we've yet to see. Bar none. We're looking forward to more MicroSource packages."

Autoscribe — The Paperwork Manager™. Here's the lowest-cost answer to your office paperwork problems. And, the only word processor that's also a small business computer. Less than $8,000 (under $200/mo. on a financing lease) with a North Star-based system (sugg. list).

Bookkeeper — The Office Accountant™. Now you can use the same system used by CPA firms to prepare your monthly statements. What used to take you days now takes just hours. Just $7595 (under $200/mo. on a financing lease) with a North Star-based system (sugg. list).

MICRO SOURCE™

1425 W. 12th Place • Tempe, Arizona 85281 • (602) 894-9247 • Cable: MICROAGE
Telex: 165 033

The following are trademarks of The Phoenix Group, Inc.: MicroSource, Autoscribe, Bookkeeper

DEALERSHIPS STILL AVAILABLE

Circle 226 on inquiry card.

Build a Computer Controlled Security System for Your Home

Steve Ciarcia
POB 582
Glastonbury CT 06033

Software consulting by Steve Sunderland

Part 1

Ciarcia's Circuit Cellar

"Steve? Lloyd? How did you get in?" I jumped back in my swivel chair in surprise as they approached.

"Joyce let us in."

It wasn't unusual for Steve and Lloyd to drop over. We've been friends and fellow computer buffs for years. The surprising part was that they hadn't seen my new basement since I had moved and they decided not to wait for an invitation.

"Boy, when you said you moved to the woods, you really meant it." I was sure that a western Long Islander like Lloyd probably considered more than three trees and a hedge in a yard a forest, but I did have to admit that the last 100 yards along my dirt road was a real killer. "How do you plan to get out of here in the winter?"

A picture of Nanook of the North complete with dogsled and team flashed into my mind. I felt the cold wind blowing in my face and frosty icicles forming in my bushy beard. Protecting my eyes from the blizzard conditions with a heavily gloved hand, I searched the horizon for the faint wisp of smoke rising from a chimney that would provide some respite from the harsh winter. The whip in my other hand was glazed with ice and no longer produced the crisp snap that was necessary to command the attention of the lead sled dog. The iciness enveloped me and began to cloud my consciousness. I could only hope that instinctual survival reactions would be triggered in time. . . .

"Steve! What are you going to do in the winter?" Lloyd repeated, breaking me from my trance.

Thoughts of my arctic ordeal instantly dissolved and the true answer came to mind. "A Jeep came with the house. Come over when it snows and I'll show you how to plow a road."

Lloyd shuddered. The thought of being bounced around in a Jeep while plowing a rutted dirt road was more than he could take. "No thanks. Maybe I'll watch — from inside!"

Before I could discuss the merits of mechanized dogsleds and their relationship to my fantasy, Steve asked a question I had been concerned about but had not yet resolved. "Have you thought of a security system for this place?"

"Actually, I've done more than think about it, I've started to design one." I was aware of my new isolation, and while crime was an important consideration, any security system I designed would do far more than just ring an alarm bell.

"Yeah! You ought to see the system I have," Lloyd piped in. "It cost a fortune, but anybody breaking a window or opening a door will set it off. It automatically dials the police too."

"Lloyd, I've been to your house. It's wired like Fort Knox! You've got foil tapes on every window like a jewelry store. This house is a contemporary, if you didn't notice as you entered. Do you realize how much glass there is in the living room alone?"

"It's a pretty foolproof method," he answered as though not hearing my question.

"Well . . . ," I continued, "there are over 300 square feet of glass in that one room. All I need is 300 square feet of strips. And didn't you say you often had false alarms in the spring and fall when the temperature variations induced cracks in the tapes?"

"Hold it Steve. Lloyd was only talking about one possibility. You don't have to install the same system. In fact there must be dozens of commercial burgler alarms you could have installed."

Steve was right. "I'm sorry guys. Moving out here in the woods has pointed out the need for self-sufficiency and independence. That's why I have the wood stove over there. I'm even going to install an auxiliary gener-

ator to power the water pump and lights. The only item I haven't settled yet is the security system."

"Isn't it a simple matter of stringing some door and window switches, an on/off key switch, and an automatic dialer?"

"Sure Steve, if I wanted a simple burglar alarm. You should know me better than that, though. Reasonably priced commercial systems are nothing more than what you've described. As soon as you add time related activations, battery backup, intrusion sensing to cover approach roads and surrounding property, and a decision making capacity from the alarm controller, dedicated logic becomes too expensive."

"What kind of system do you have in mind?" inquired Lloyd.

"Actually I haven't finalized all the details, but the system I want would do things like turn on the outside lights as I drive up to the house, activate 115 VAC appliances in either a preprogrammed sequence or randomly, have display panels in the bedroom and down here in the basement to track anyone approaching the house, and of course all the usual fire and burglar alarm functions."

"Wow, Steve!" Lloyd's eyebrows rose a little as the security system was unveiled. "That would take a computer!"

"Exactly!" I exclaimed.

Steve looked at me, then at the big (by microcomputer standards) computer to my left. His technical mind did a quick calculation on the battery backup requirements for my 64 K dual disk computer system.

"I suppose you've laid in a separate power feeder from the Grand Cooley Dam to keep this dinosaur running?" he said.

"Don't be funny. The computer I propose to use is a single board microcomputer, programmed to provide the sophistication I want but versatile enough to allow logic changes and easy hardware expansion." I reached into the drawer behind me and pulled out a foot square populated printed circuit board and passed it to Steve. "Here's the computer." Both Steve and Lloyd eyed the odd looking

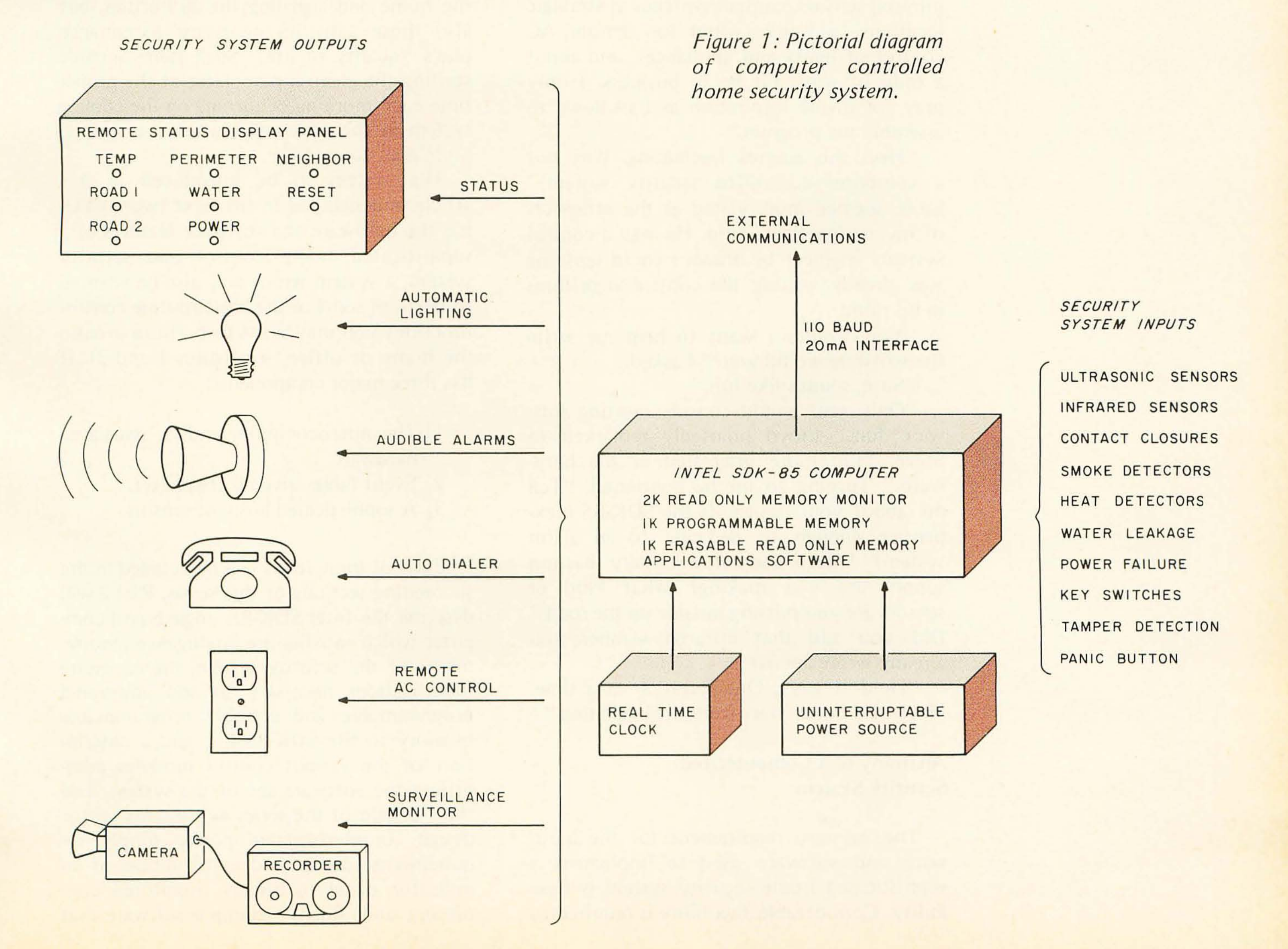


Figure 1: Pictorial diagram of computer controlled home security system.

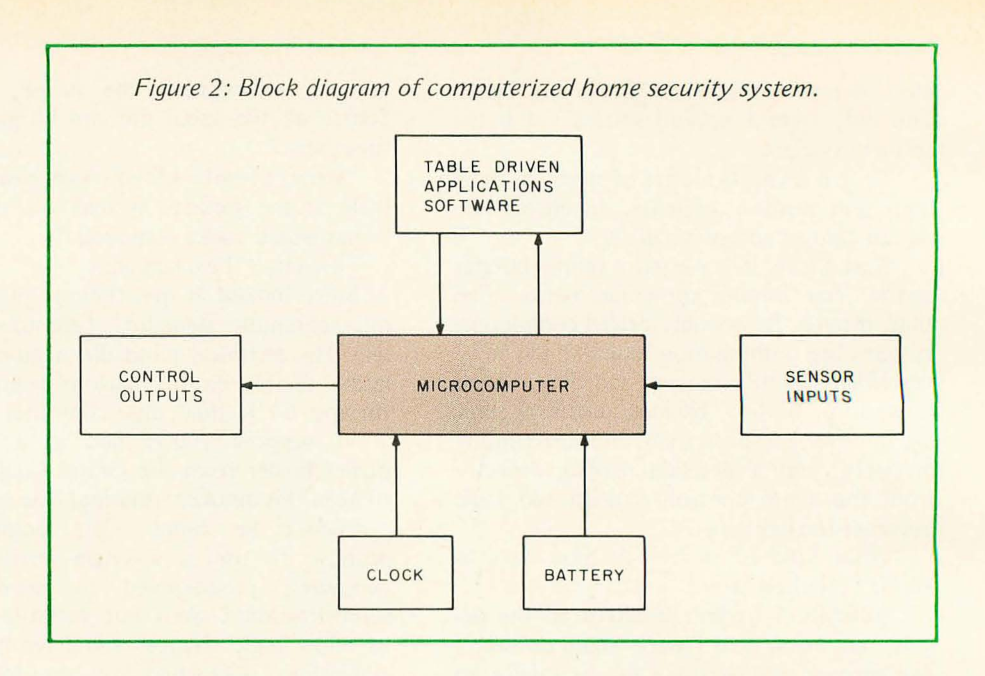


Figure 2: Block diagram of computerized home security system.

circuit. "It's an Intel SDK-85 kit which uses an 8085 processor. With a little additional programmable read only memory to store the program, some ultrasonic and infrared sensors, contact switches at strategic locations, a house wired for remote AC control of lights and appliances, and about a mile of wire, I'll be in business. I only pray for divine inspiration as I sit down to assemble the program."

"Hey, this sounds fascinating. Why *not* a computer controlled security system!" Steve seemed more elated at the prospects of my design than I did. He was a control systems engineer by trade; I could sense he was already writing the control algorithms in his mind.

"You wouldn't want to help me write the software would you?" I asked.

"Sure, sounds like fun."

"Only you would consider writing software fun." Lloyd pointedly remarked to Steve. "I'm more interested in the hardware." Turning to me he continued. "Tell me about your design. Is the SDK-85 inexpensive enough to dedicate to an alarm system? What kind of a battery backup supply are you making? What kind of sensors are you putting outside on the roads? Did you add that infrared scanner you already wrote about?"

"Hold it guys. One question at a time. Maybe I'd better start from the beginning."

Anatomy of a Computerized Security System

The primary requirement for the hardware and software used to implement a sophisticated home security system is flexibility. Considerable flexibility is required to allow you to structure a system which executes a variety of functions. These functions include not only the menial tasks of notifying residents that an intruder has entered the home and signaling the authorities, but also those activities necessary to enhance one's "quality of life." Such items include starting the electric percolator at the proper time each morning or turning on the cooling system in the wine cellar when the temperature goes above 64°F.

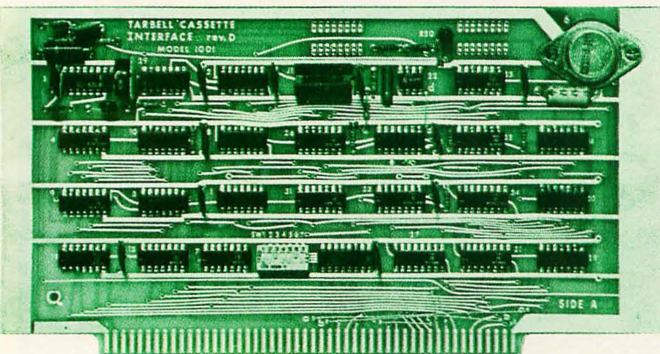
The system to be introduced in this article and detailed in the next two BYTEs has the hardware and software features of a sophisticated home control and security system, a system which may also be adapted to perform some of the basic routine control functions you may want to perform around the home or office (see figures 1 and 2). It has three major components:

1) The microcomputer and its associated hardware.
2) Event-table-driven software set.
3) A sophisticated array of sensors.

Each of these items will be detailed in the succeeding sections of this series. Part 2 will describe the Intel SDK-85 single board computer which satisfies the intelligence requirements of the security system, the hardware modifications necessary to add additional programmable and erasable programmable memory to the SDK-85 kit, and a description of the various control modules comprising the software set of the system. The third article of the series will discuss sensor design, uninterruptable power supply requirements, design and construction of an indicator panel to display monitored conditions, and, finally, example software that

TARBELL SETS STANDARDS
For Hobbyists and Systems Developers

Sales to thousands of hobbyists over the past two years have proven the Tarbell Cassette Interface to be a microcomputer industry standard. Tarbell Electronics continues research and development to produce new and efficient components to fill hobbyists' changing needs.

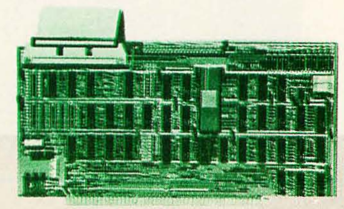


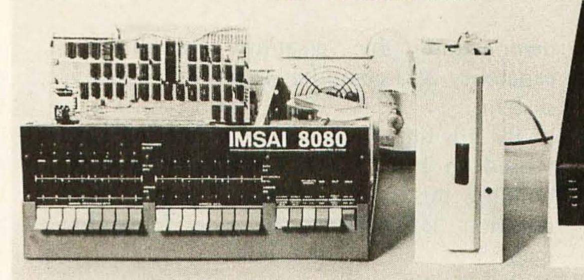
TARBELL CASSETTE INTERFACE

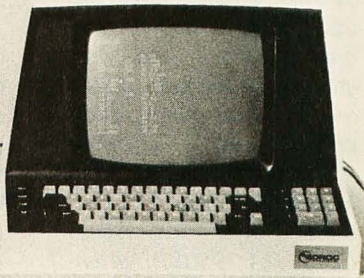
- Plugs directly into your IMSAI or ALTAIR*
- Fastest transfer rate: 187 (standard) to 540 bytes/second
- Extremely Reliable—Phase encoded (self-clocking)
- 4 Extra Status Lines, 4 Extra Control Lines
- 37-page manual included
- Device Code Selectable by DIP-switch
- Capable of Generating Kansas City tapes also
- No modification required on audio cassette recorder
- Complete kit $120, Assembled $175, Manual $4
- Full 6 month warranty on kit and assembled units

TARBELL FLOPPY DISC INTERFACE

- Plugs directly into your IMSAI or ALTAIR* and handles up to 4 standard single drives in daisy-chain.
- Operates at standard 250K bits per second on normal disc format capacity of 256K bytes.
- Works with modified CP/M* Operating System and BASIC-E Compiler.
- Hardware includes 4 extra IC slots, built-in phantom bootstrap and on-board crystal clock. Uses WD 1771 LSI Chip.
- Full 6-month warranty and extensive documentation.
- PRICE:
  Kit $190 ...... Assembled $265





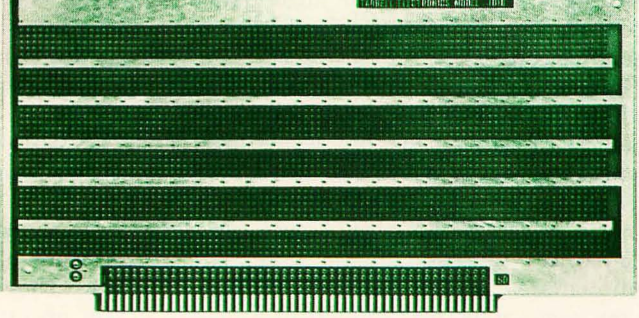


CP/M with BASIC-E and manuals: $100

Compatible Disc Drives

Ask about our disc drives priced as low as $525.

TARBELL PROTOTYPE BOARD
Model 1010



- Gold plated edge pins
- Takes 33 14-pin ICs or
- Mix 40-pin, 18-pin, 16-pin and 14-pin ICs
- Location for 5 volt regulator
- Suitable for solder and wire wrap
- ALTAIR/IMSAI compatible

Price: $28.00

For fast, off the shelf delivery, all Tarbell Electronics products may be purchased from computer store dealers across the country. Or write Tarbell Electronics direct for complete information.

*ALTAIR is a trademark/tradename of MITS, Inc.
CP/M is a trademark/tradename of Digital Research



950 DOVLEN PLACE • SUITE B • CARSON, CALIFORNIA 90746
(213) 538-4251 • (213) 538-2254

Circle 360 on inquiry card.

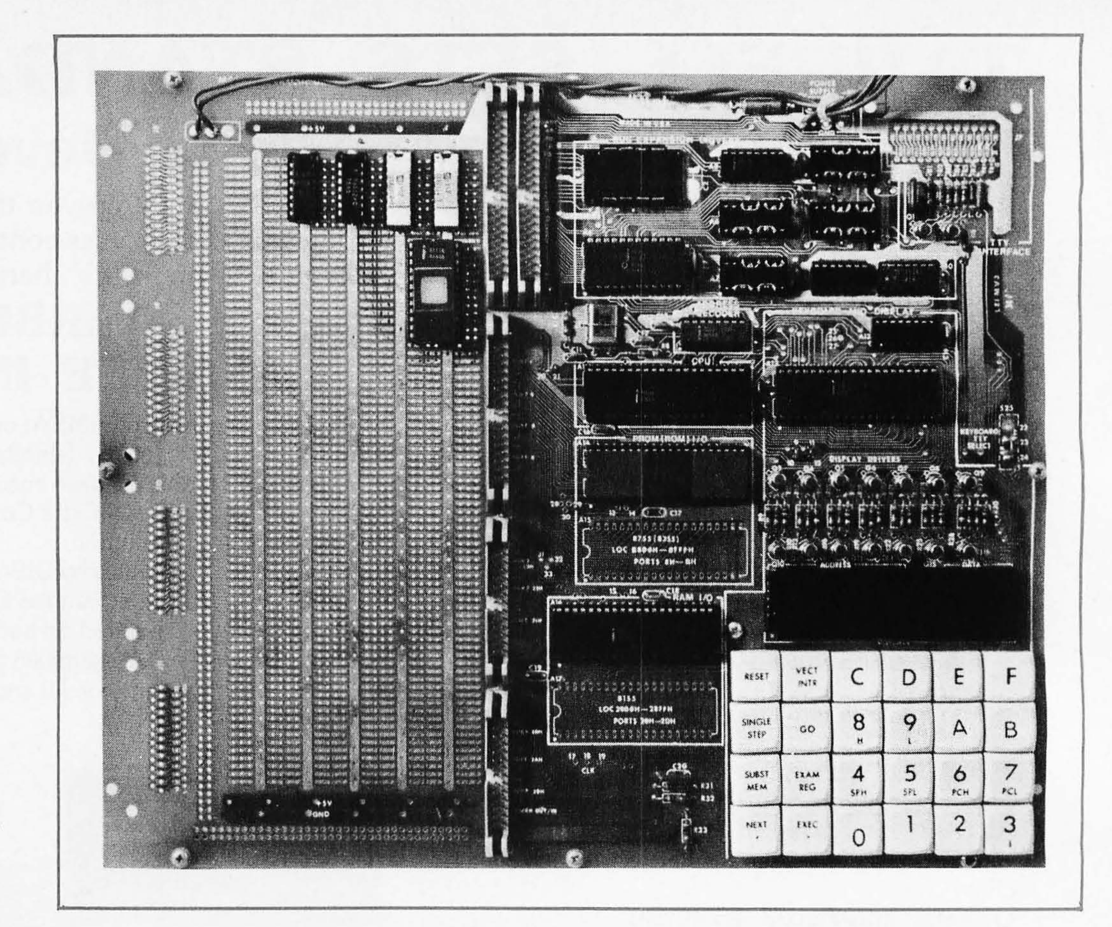


Photo 1: The Intel SDK-85 single board computer, that forms the heart of the computer controlled security system described in this article. Note the additional erasable read only memory and programmable memory added in the prototyping section at upper left. For more information about the SDK-85, contact Intel, 3065 Bowers Av, Santa Clara CA 95051, attention: Rob Walker.

demonstrates the versatility and ultimate capability of a computer controlled security system.

By this time you may be wondering why I used an Intel SDK-85 microcomputer in building this system. Before I answer this question, let's discuss the general requirements.

A computer home security system's link with the real world is its sensors. The sensing devices must be small, reliable, low power, and capable of detecting exceptional sound, light and motion. No single device I used was capable of detecting changes in these three conditions, with the exception of the infrared scanner described in a previous article (see "I've Got You in My Scanner! A Computer Controlled Stepper Motor Light Scanner," by Steve Ciarcia, November 1978 BYTE, page 76). A variety of single function sensors provided the necessary inputs: ultrasonic devices to sense motion within the monitored area, pressure switches to monitor the removal or the addition of a mass (such as a foot on the stairs to the Circuit Cellar), photoelectric devices to detect movement between monitored points, and an array of contact closures to signal the opening and closing of strategic doors and windows.

All inputs to the computer are designed to be discrete in nature. That is, they exist as switched outputs and appear to the computer as either a high or low logic level in the activated state, depending on the particular sensor. The majority of the sensors have relay outputs and can be wired in series or parallel combinations to cover wide areas or to give redundant indication. An illustrative example is using the series combination of an ultrasonic and infrared sensor to cover large open areas. The failure or false indication of a single input will not cause a false alarm, since both sensors must have a positive "intruder" presence. In effect, the signals from the two devices are logically ANDed. More on these techniques when we discuss sensors.

The Alarm Contains More Than a Switch and a Bell

Functions other than the detection of intrusion have been incorporated in the system. These functions are centered on the safe operation of the home or office, and are used to detect the status of sump pumps, refrigeration systems and water pumps. Failures of these devices are detected by the system again via simple contact closures. The detection of smoke and fire has been incorporated into the system for the protection of the property and its occupants.

Figure 3 is the floor plan of a contem-

introducing . . .

YOUR NEW LIFE STYLE!



Discover How to Improve Your Personal Abilities

An amazingly powerful application of a computerized personal information management system. In handy booklet form.

- Increase your personal capabilities
- Save money
- Improve your ability to plan
- Locate important facts quickly
- Eliminate the drudgery of routine chores

These all add up to a better life style for you! A new way of living which can be more pleasant, bring you more happiness and success.

Information. That's the key. Your command of information is what gives you the power to succeed.

Information For What?

Information you need to make decisions . . . to solve problems . . . to seek creative solutions in real life situations. Information that when under your control will amplify your personal abilities.

You've Heard of the Information Explosion?

Now it's time for an explosion in your personal capacity to deal with a wealth of information. The personal computer is the answer. And now the means is available for you to use it to your personal advantage.

Can a Computer Really Do It?

You bet it can! The power of a personal computer is fantastic. Untold amounts of information can be processed in the blinking of an eye. Chores you've found laborious can be done by your simple command. Facts that have taken hours to find can now be located at the snap of your finger. That's what a personal computer can do . . .

. . . if YOU know how!

YOU? Yes, you! You need to have the know-how to unleash the power of the microcomputer for your benefit.

Until now only the programmers and designers could make such profitable use of the personal computer. PIMS — the Personal Information Management System — has changed that. You don't have to be a programmer to use PIMS. PIMS is a program which you type into a personal computer. Then it's just a matter of following the instructions for setting up your own tasks for computer assistance.

Can It Really Be That Easy?

Sure. That's the beauty of a program for information management. It's an alternative to programming from your standpoint. All you do is define the job you want the microcomputer to do. Express yourself in simple commands and statements. Then the microcomputer plus PIMS does the rest.

What Can It Do?

PIMS will enable you to function with increased effectiveness. Use it to balance your check book. It can keep an inventory of your possessions. (Ever store something and afterwards forget where? PIMS will tell you the location!) Use PIMS to plan your day, vacation, education and important life events. PIMS can tutor math, keep your personal mailing list or telephone directory. Use it to keep track of personal disbursements. It can even be used to schedule your TV viewing. And more. Much more. There's practically no limit to the type of applications.

Here's what you get.

All in one neatly bound volume,

- The microcomputer and its potential for personal use
- Ways in which your microcomputer can be helpful in everyday life and serving basic personal needs
- PIMS — Personal Information Management System. All the basics and jargon
- PIMS program outline and flow chart
- 15 sample ways in which PIMS can work to your personal advantage
- The complete listing of a practical functional program. Written in BASIC language. Ready to use!

The PIMS program was prepared by SCELBI using Microsoft† compatible BASIC as used in a wide variety of personal computers. Systems like Apple II, Radio Shack TRS-80 (level II), Ohio Scientific 400 and many other small computer systems.

Don't Have a Computer?

Even if you are still just thinking about getting your own personal computer, you should have this book. It will show you in clear understandable terms what you'll be able to accomplish with your own microcomputer.

Is the Message Clear?

PIMS — Personal Information Management System can help you. If you're interested in understanding what role a computer can play in enhancing your life style, get PIMS. It will give you the inside track for getting the most out of a personal computer system.

Get Yours!

PIMS is available in handy booklet form. Your copy is ready for shipment. At only $9.95* it's got to be the greatest bargain to come along!

Don't Wait.

Get your copy now! You'll discover how you can find your new life style — with PIMS.

*Prices shown are for North American customers. Pricing, specifications, availability subject to change without notice. Personal checks may delay shipment. IMPORTANT! Include 75¢ postage/handling for each item to be delivered by U.S. Mail Book Rate, or $2 for each item to be shipped First Class or UPS.

† A trade mark of Microsoft, Inc.

SCELBI Publications
P.O. Box 133 PP STN
Milford, CT 06460

YES! I want PIMS.
Enclosed is $9.95 plus 75¢ postage/handling. Or, better yet, please charge my Master Charge or VISA account:

Card No.__________
Exp. Date__________
Bank No.__________
Name__________
Address__________
City__________
State__________ Zip__________
Signature__________

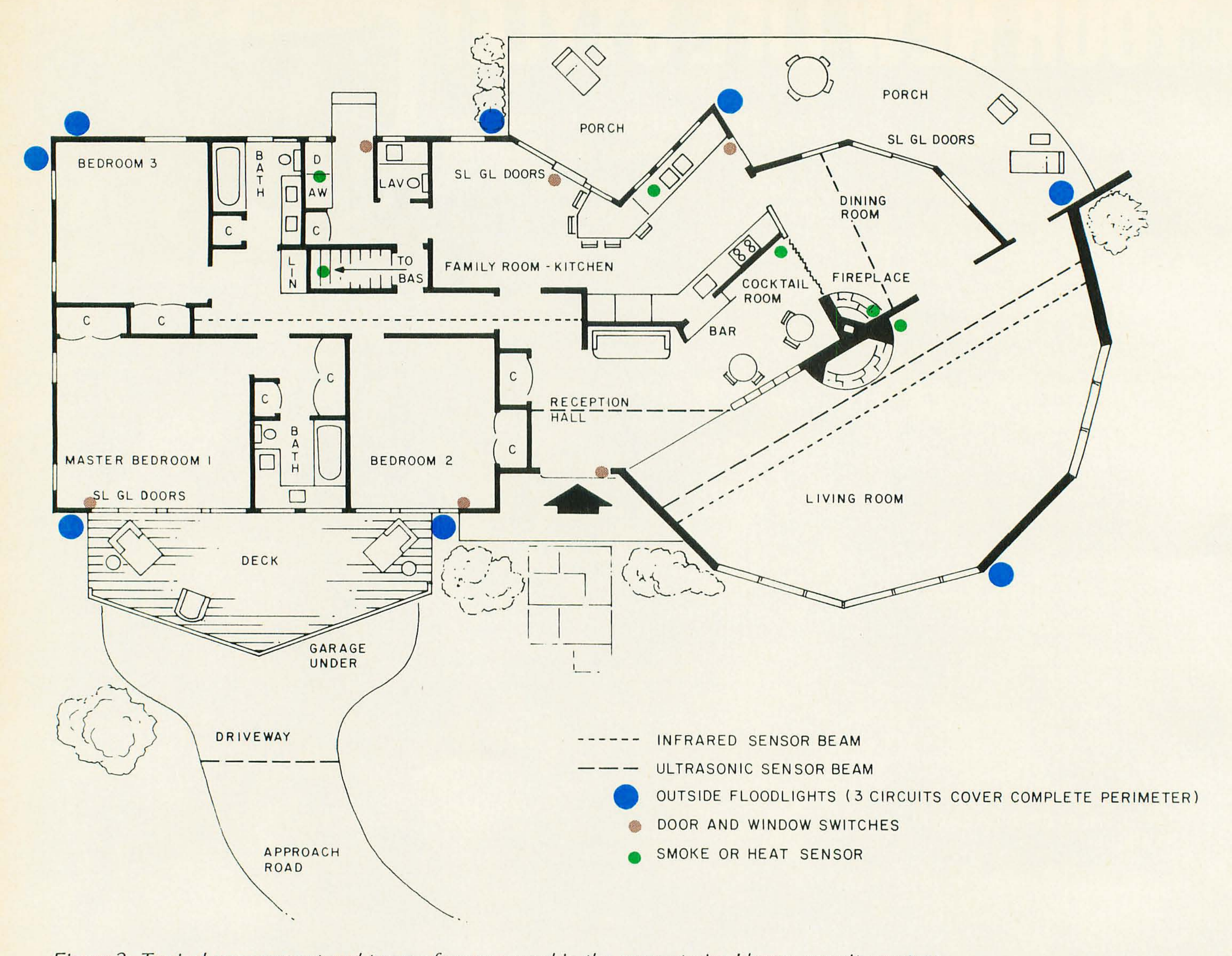


Figure 3: Typical arrangement and types of sensors used in the computerized home security system.

porary house similar to my own. For obvious reasons I have chosen not to use one of my own home. The dimensions and layout are similar enough to adequately represent the model and to illustrate the possible quantity, type and placement of sensors. I consider my house a unique application and do not believe that every detail of this system is necessary for adequate security coverage. Less sophisticated photoelectric and ultrasonic sensor designs could be used to monitor a 9 by 12 foot room, but might be inadequate in a much larger room. My home proved to be a sensor development challenge, since the living room is 42 feet in diameter and the house totals 4800 square feet monitored area. The best procedure to follow is to understand the theory of the system, differentiate between the uniqueness of my application and a general case, and extract those components which satisfy your security needs.

Alarm Priorities

The inputs monitored by this security system are divided into four priority levels, level 0 through level 3:

Level 0

Activation

Level 0 alarms are always enabled as long as the system is operational. Inputs of this type consist of smoke sensors, heat activated switches, and panic button inputs.

Result

Positive closure of any sensor immediately triggers an audible alarm within the residence. Automatic dialing of police, fire and neighbor commences 45 seconds later. This delay allows time to reset system in cases of false alarms, such as burned toast.

Level 1

Activation

Level 1 alarms are enabled by a keylock switch input to the computer and are primarily intrusion in nature. They consist of parallel and series combinations of ultrasonic, infrared, and contact closure type sensor inputs.

Result

Positive closure of any sensor immediately triggers an audible alarm inside and outside. The system immediately dials police, contacts the neighbors via a dedicated serial communication link between houses, and turns on surveillance recording devices.

Level 2

Activation

Provision is made to allow entry to the house at selected spots to deactivate the alarm. Entering these points starts a timer which must be reset by deactivating the level 1 alarm system.

Result

Failure to reset the system within the alloted time triggers a level 1 condition with typical level 1 response.

Level 3

Activation

Level 3 is always activated and its input sensors are displayed on a panel as well as being supplied to the computer for possible action. Sensor inputs include power failure, perimeter intrusion, freezer failure, etc.

Result

Perimeter or approach information can be utilized by the system to turn on outside lights, but the primary purpose of level 3 is to provide a noncritical condition monitoring system for information. The duration of the displayed event is a timed function.

The combination of the four levels uses a total of 20 parallel input bits on the SDK-85.

Alarm Outputs

Unlike the majority of security alarms, which have only a single output to turn on the automatic dialer and alarm bell, this system has 18 discrete output functions. They can be activated in any combination, and are described in table 1.

The particular designations for inputs and outputs are selected for my needs, and are easily redefined due to the program structure and event table software. A pictorial diagram is shown in figure 1. This will be explained in detail next month.

Additional Considerations and Capabilities

Another major consideration in a home security system is the provision of a continuous and reliable power source. This is necessary to sustain operation during electrical interruptions. To assure a dependable power source, the system is powered by an uninterruptable power supply. Necessary input voltage is provided, at all times, by 12 V batteries. The charge on the batteries is maintained by an automatic 115 VAC charger which will charge the batteries when their output voltage falls below a preset value. The output of the power source is used to supply voltage to critical sensors, the status display panel, and the microcomputer. The computer will of course register the occurrence of a power outage.

As stated previously, one capability of this system is the controlled activation of AC outputs triggered by input events or timer generated commands. To accomplish the latter, one needs a time base and a real time clock. The circuit in figure 4 is added to the SDK-85. The real time clock creates a pulsed output with a frequency of 1 Hz. This pulse output signal, when attached to the interrupt

Table 1: 18 discrete functions performed by the author's computer controlled security alarm system.

Output	Function
1. Auto dialer	Automatically calls police, fire, etc.
2. Audible alarm (intrusion)	Signals that there has been a break-in.
3. Audible alarm (fire)	Signals detection of fire or smoke.
4. Level 1 enabled (light next to key switch)	Steady output indicates reset condition; flashing output indicates enabled mode.
5. Perimeter flood lights	Activated by timer or level 3 input.
6. Driveway flood lights	Timer activated at dusk.
7. Low level audible alarm	Signals pertinent combination of events or particular level 3 activation; mounted in level 3 display panel.
8. Recording instruments	Particular level 1 alarms trigger activation of surveillance recording devices.
9., 10., 11. 3 bit, level 3, display board driver	Multiplexed output controls eight display parameters.
12. AC circuit 1 13. AC circuit 2 14. AC circuit 3 15. AC circuit 4 16. AC circuit 5 17. AC circuit 6	Six AC power control outputs to lights or appliances which either simulate occupancy through sequential activation or act as remotely controlled outputs to turn on percolators, air conditioners, etc.
18. Output to neighbor	Indicates fire or intrusion.

"My Structured Systems business software has paid for itself in labor hours saved alone."

Mr. Ken Tunnah, Colloid-A-Tron Inc., Buffalo, New York



Ken Tunnah is one of many innovators bringing the micro revolution to the small business. As a programmer, he knows computers and their languages. As a businessman, he knows business and its languages. And when Mr. Tunnah decided to microcomputerize the accounting function at Colloid-A-Tron, he turned to Structured Systems software.

Says Mr. Tunnah: ***"The program is designed from a CPA standpoint, for multiple corporations, which we have. It is flexible and gives me the ability to change reporting by profit centers easily. It is up and running quickly, and it just keeps on running. I think it's the best business software available."***

The best software available. That's what Structured Systems Group set out to create.

Structured Systems offers three sophisticated accounting systems. Our General Ledger software is big enough for multi-client write-up by the CPA, or multi-corporate reporting for the business, but small enough for the micro budget. The very comprehensive Accounts Receivable and Accounts Payable packages will operate independently, or they will coordinate with the General Ledger.

Our systems record transactions easily and correctly, and provide an audit trail from source document to financial statements. And they will maintain monthly and year-to-date information in dollars and in percentages. And they are reliable.

The three systems interact with the user to set up parameters such as format and headings, account titles and numbering, automatic billing or reminder notices, credit limits, sales reports, a check register, and much more.

The software is designed to run on an 8080 or Z-80 CPU with 48K of memory, dual disks with CP/M®, printer, keyboard, and CRT. To make it all work for you, we have provided the most extensive documentation and support in the industry.

We provide the capability to computerize complex accounting functions on relatively inexpensive microcomputer equipment. Ken Tunnah has told us what that means: ***"I've bucked some trends. I looked around, and decided that with the right software, I could get a micro to outperform a $45,000 mini. I'm satisfied. It's simple economics."***

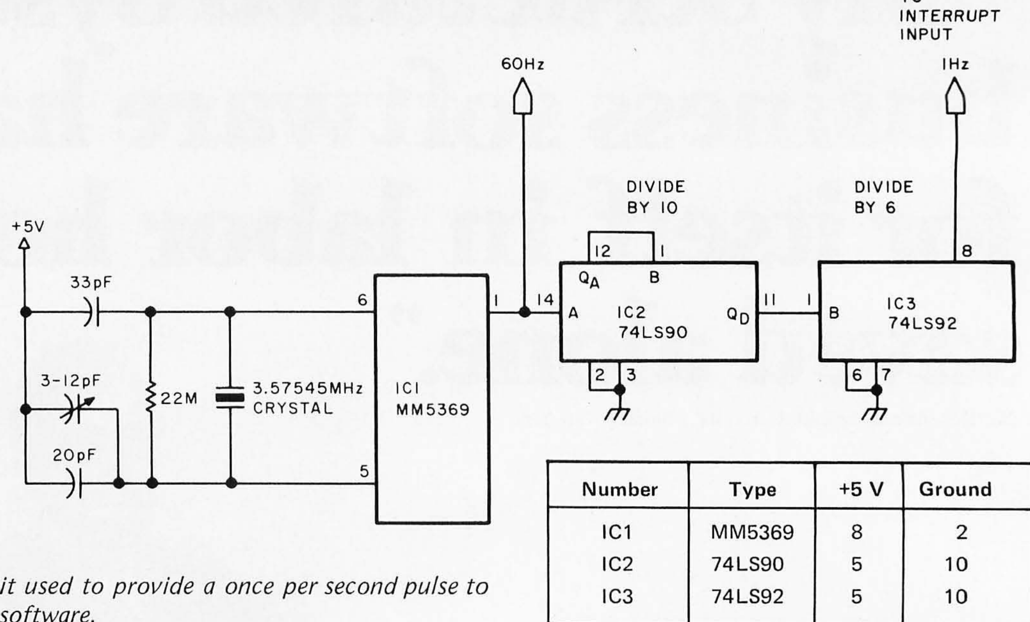
We can refer you to a growing number of sophisticated retailers experienced in Structured Systems Group business systems. Or we can work directly with you. We'd be happy to provide you with more information on our product line, which includes QSORT™ (a sort/merge program), CBASIC (a business BASIC), NAD™ (a mailing and addressing system), and our General Ledger, Accounts Receivable, and Accounts Payable packages.

®CP/M is a registered trademark of Digital Research

ACCOUNTS PAYABLE
GENERAL LEDGER
ACCOUNTS RECEIVABLE

Structured Systems Group
INCORPORATED
5208 Claremont Avenue
Oakland, CA 94618
Call us at (415) 547-1567. Or write Dept. B7.

Putting the Microcomputer in Business.



Number	Type	+5 V	Ground
IC1	MM5369	8	2
IC2	74LS90	5	10
IC3	74LS92	5	10

Figure 4: Real time clock circuit used to provide a once per second pulse to update the SDK-85 time of day software.

input line of the 8085 and combined with proper software, provides the alarm system with a time of day clock having resolution of 1 second. The current time in hours, minutes and seconds is constantly displayed and updated on the 6 digit LED display on the SDK-85 board. This gives a constant indication of system operation and readiness. This display can be deactivated to conserve power during a power interruption.

The time of day clock is not the only operator interface. An additional display panel is remotely located in the bedroom or similar area where current system status can be viewed. This display is not meant to indicate the obvious, but rather to make the resident aware of less critical but equally important matters. In the event of a fire or outright break-in, the system would activate the necessary counter measures and notify the authorities. Flashing a tiny light emitting diode (LED) on a panel will hardly be noticed in all the noise of the sirens.

A third level of sensors monitors approaches to the house, the perimeter, and important events such as water leaks and power failures. While the system may still turn on outside lights if desired, this panel would immediately notify the resident of someone passing through the property, or of a power failure. When you live in the woods, these are all important considerations.

Photo 2: Prebuilt commercial ultrasonic transmitter (small board) and receiver available from Bullet Electronics; for use across doorways, rooms, etc.

Using an Intel SDK-85

The microcomputer system utilized in this security system was constructed around an Intel MCS-85 system design kit (SDK-85). It was chosen because of the basic features it provides to the experimenter. Included in this $250 kit are all the materials necessary to build a basic computer system which can be modified easily with additional programmable read only memory containing the security system software. The main features provided in the standard SDK-85 include:

- 8085 processor utilizing the 8080A instruction set.
- Interactive 6 digit LED display and 20 key keyboard.
- 110 baud 20 mA current loop interface.

the $988 Surprise . . .

If you haven't looked carefully at the Level-II 16K TRS-80, you're in for a big surprise! Level-II BASIC gives TRS-80 advanced features like comprehensive string handling, multi-dimension arrays, multi-letter variable names, named cassette files, full editing, integer arithmetic, single (6-digit) and double (16-digit) precision arithmetic, formatted printing, memory-mapped video (print directly at any of 1024 screen positions), 128x48 video graphics (may be intermixed with text), error trapping, auto line numbering, TRACE, PEEK and POKE . . . to name just a few. Because Level-II is in ROM, TRS-80 powers-up ready to go with the full 16K RAM available for your use.

This means TRS-80's memory is equivalent to a 28K RAM-based system.

New for 1979—TRS-80's numeric (calculator) keypad included on every 16K computer, and available as an add-on for present owners.

TRS-80's modular design allows easy expansion. Add up to 48K RAM, Expansion Interface, printers, 1 to 4 Mini-Disks, RS232C, telephone acoustic couplers, Voice Synthesizer, dual cassette recorders, our System Desk and Printer Stand. Surprisingly, these are not promises of things to come, but real products being delivered right now. Software from games to General Ledger are available, with more cassette and disk software being added monthly.

Radio Shack's 58 years of consumer electronics leadership, our 50 regional repair centers (growing to 100 this year), our new Radio Shack computer centers, and our NYSE-listed billion-dollar parent, Tandy Corporation, insure that customer support is always available right where it should be—locally.

So if you haven't seriously looked at TRS-80 yet, ask your local Radio Shack for our new 20-page fact-filled catalog and be prepared for a $988 surprise. Surprising power—features—price—support! Level-II 16K systems include everything pictured, plus the manual. Better to be surprised now . . . before you choose the wrong microcomputer system.



- Comprehensive read only memory based monitor (2 K bytes).
- 256 bytes of programmable memory.
- 38 parallel input output (IO) lines.
- Three user-vectored interrupts.

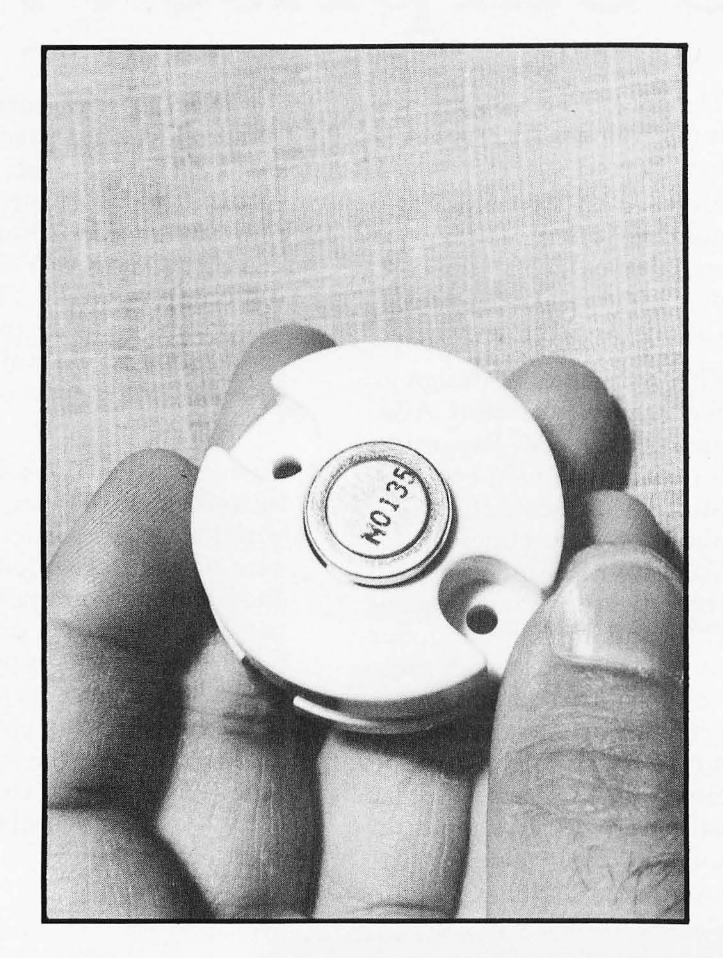


Photo 3: Heat sensor designed to trip above 135° F, typical of the type often used in combination with smoke detectors.



Photo 4: Typical commercially available sensors. Shown are a panic button (left) and magnetic door switch, available from GC Electronics, Rockford IL, and Radio Shack, respectively.

The presence of a good monitor, a quantity of IO bits, vectored interrupt capability, and an interactive keyboard and display were the primary reasons for the choice of this unit. As previously stated, the real time clock interrupts the system every second. While interrupt capability is not absolutely necessary to the design of a security system, it does insure that events which may become critical to the performance of a system are not missed. For example, let's assume that maintaining an accurate time base is required, and that the system maintains time by a pulsed input. If this input is interfaced to the computer via a line on an input port, it may be possible at times (depending on processor load) to miss the event, thus causing the system to slowly lose its time synchronization to the real world. However, if this input is interfaced to the system via an interrupt line, the system will not "miss ticks" due to processor overhead. Should the processor be busy at the time the interrupt occurs, two possible conditions could exist. First, the routine being executed could have temporarily inhibited interrupts. This will only delay the processing of the timer interrupt: when the interrupts are enabled by the routine the processor will recognize that the interrupt is pending and vector to the appropriate address to begin processing the interrupt. Note that, should interrupts be inhibited for a period longer than the frequency of the interrupt, "missed tick" conditions will occur. Therefore, one should always be aware of the maximum time during which interrupts will be inhibited. The second condition which can arise during the processing of interrupts is that the processor might be busy executing a module with the interrupts enabled. In this case the processor will immediately vector to the routine to process the interrupt. Upon completion of the interrupt processing, control is returned to the interrupted module.

The keyboard and display are important to the operation of the security system. These units are utilized extensively in the startup of the system.

It is through this facility that the user initiates the operation of the system, notifies the computer of the current time of day so that it may initialize the software clock, and can utilize the monitor to diagnose problems within the system. For example, the key-

Introducing the simple TRS-80 Up-grade

Fast, easy, guaranteed expansion to 16K at less than half the price of Radio Shack.



Only $140

Ithaca Audio makes it simple

No false starts and finding you need some little item or special tool. Our Kit contains all the parts: 8 prime dynamic RAMs and a complete set of preprogrammed jumpers. No matter which model you have (even if you later purchase Level II software), you're covered.

Complete Instructions

Our easy-to-follow directions cut installation time to just minutes. You can do it yourself—with no soldering! All you need is a household screwdriver.

100% Guarantee

Like our kit, simple: if a part ever fails, we replace it, FREE.

Available now

Order from your favorite retailer. If by chance he hasn't stocked them yet we'll ship him your Kit right away.

For technical assistance call or write to:

ITHACA AUDIO

Phone: 607/273-3271
P.O. Box 91 Ithaca, New York 14850

Available off-the-shelf at these fine computer dealers.

AL: BIRMINGHAM: Computer Center, (205) 942-8567. HUNTSVILLE: Computerland, (205) 539-1200. **CA:** BERKELEY: Byte Shop, (415) 845-6366. EL CERRITO: Computerland, (415) 233-5010. HAYWARD: Computerland, (415) 538-8080. LOS ALTOS: Computerland, (415) 941-8154. MARINA DEL REY: Base 2, (213) 822-4499. MT. VIEW: Digital Deli, (415) 961-2670. SAN FRANCISCO: Computerland, (415) 536-1592. SAN JOSE: Electronic Systems, (408) 226-4064. SAN RAFAEL: Computer Demo Room Inc., (415) 457-9311. WALNUT CREEK: Computerland, (415) 935-6502. **DE:** NEWARK: Computerland, (303) 738-9656. **FL:** FT. LAUDERDALE: Computer Age, (305) 791-8080. POMPANO BEACH: Computer Age, (305) 496-4999. TAMPA: Micro-computer Systems, (813) 879-4301. **IL:** NILES: Computerland, (312) 967-1714. OAK LAWN: Computerland, (312) 422-8080. PEORIA: Computerland, (309) 688-6252. **KS:** OVERLAND PARK: Personal Computer Center, (913) 649-5942. WICHITA: Computer Systems Design, (316) 265-1120. **KY:** LOUISVILLE: Computerland, (502) 425-8308. **MA:** CAMBRIDGE: Computer Shop, (617) 661-2670. **MD:** ROCKVILLE: Computerland, (301) 948-7676. **MI:** ANN ARBOR: Newman Computer Exchange, (313) 994-3200. ROYAL OAK: Computer Mart, (313) 576-0900. **NJ:** ANDOVER: Atlantic Microsystems, (201) 549-0189. BUDD LAKE: Computer Lab of New Jersey, (201) 691-1984. CLARK: S-100, (201) 382-1318. ISELIN: Computer Mart (201) 283-0600. SUCCASUNNA: Computer Hut, (201) 584-4977. **NY:** BUFFALO: Computerland, (716) 836-6511. ITHACA: Computerland of Ithaca, (607) 277-4888. JOHNSON CITY: Micro World, (607) 798-9800. NEW YORK CITY: Computer Mart of New York, (212) 686-7923. SYRACUSE: Computer Shop of Syracuse Inc., (315) 446-1284. **OH:** CINCINNATI: Digital Design, (513) 561-6733. DAYTON: Computer Solutions, (513) 223-2348. **OK:** OKLAHOMA CITY: Micronics, (405) 942-8152. **PA:** FRAZER: Personal Computer Corp., (215) 647-8463. STATE COLLEGE: Microcomputer Products Inc., (814) 238-7711. **TN:** KNOXVILLE: Eastern Microcomputer, (615) 584-8365. **TX:** AUSTIN: Computerland, (512) 452-5701. DALLAS: KA Electronic Sales, (214) 634-7870. GARLAND: Digital Research Corp., (214) 271-2461. HOUSTON: Houston Computer Mart, (713) 649-4188. **UT:** OREM: Johnson Computer Electronics, (801) 224-5361. **VA:** ALEXANDRIA: Computers Plus, (703) 751-5656. ARLINGTON: Arlington Electronics Wholesalers, (703) 524-2412. **VT:** ESSEX JUNCTION: Computer Mart of Vermont, (802) 879-1683. **CANADA:** ONTARIO: MISSISSAUGA: Arisia Microsystems, (416) 274-6033. TORONTO: Computer Mart Ltd., (416) 484-9708. WINNIPEG: Patrick Computer Systems Inc., (204) 774-1655. **WEST GERMANY:** MUNICH: ABC Computer Shop, Schellingstrasse 33, 8000 Munchen 40 Microcomputer Shop, Toelzerstr, 8, D-815 Holzkirchen. **ISRAEL:** HAIFA: Microcomputer Eng Ltd., **Haifa 31-070.**

board and display allow the user to initialize the system by:

- Resetting the processor by depressing the Reset button.
- Entering the 4 digit address of the start of the security system cold start module, which will be displayed, as entered, on the display.
- Depressing the Go button.

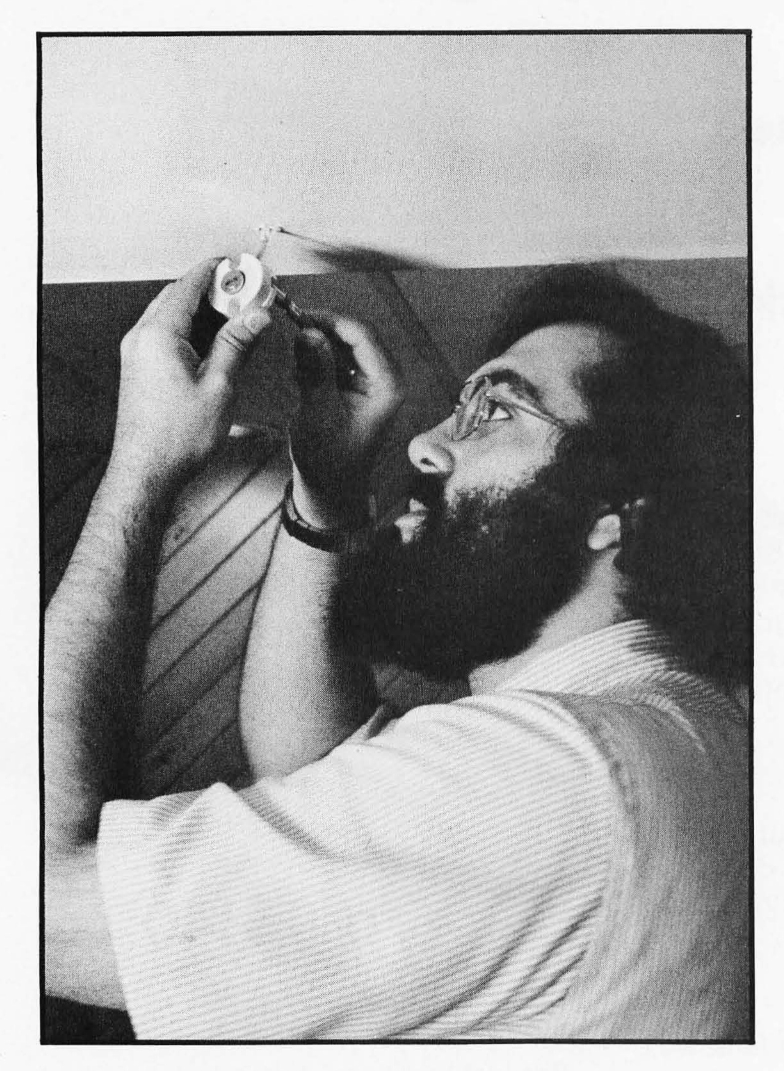


Photo 5: Steve Ciarcia installing a heat sensor over the printer and the Selectric in the circuit cellar.

The functions provided for debugging the system include:

- The ability to examine the contents of memory locations registers both randomly and sequentially.
- The ability to load new values into memory and registers both randomly and sequentially.
- The ability to single step the execution of a software module.
- The ability to generate vectored interrupts from the keyboard.
- The ability to initiate execution from various addresses.

The monitor provided in the SDK-85 resides in 2 K bytes of read only memory. It contains the software required to service the commands the processor receives from the keyboard, the routines to drive the 6 character LED display, the algorithms to process and serve Teletype (or other serial IO device) requests and the software to handle serial IO communication. This monitor also provides the vectored interrupt trap locations which in turn direct control to user-supplied modules.

Several of the monitor routines are very useful to the operator and were used extensively to display input data and do the initial debugging.

Four modules are provided for sending and receiving data to or from the Teletype. These are:

- Console input which returns an ASCII character received from the Teletype to the requesting module.
- Console output which transmits an

ASCII character from the sending module to the Teletype.

- Carriage return/line feed which outputs to the Teletype a carriage return followed by a line feed.
- Hexadecimal number printer which converts a byte of data to two ASCII characters (hexadecimal 0 thru F) and outputs them to the Teletype.

Three modules are provided for sending and receiving data to the keyboard/LED display. They are:

- Update data which displays the contents of register A on the data field of the display in hexadecimal notation.
- Read keyboard which returns to the requesting module the character entered on the keyboard.
- Output data which outputs data to the address or data field of the display.

The security system software is comprised of a set of highly structured modules (see figure 5) used to process records passed from one module to another. Within this structure the initiating event can come from one of two sources, the detection of an external event or the occurrence of a time event. The occurrence of an external event is detected by the digital scan module; the occurrence of a time event is detected by the timer interrupt processor. When either of these modules detects the occurrence of an event that requires further processing, it passes an event index to the event processor, which will initiate the processing of the event. The event processor utilizes the event index to extract the response records associated with the event in question from the event record. It then transmits this data to the response processor, which will perform the required response (ie: output a contact closure to turn on a siren or autodialer, or activate a special application module). The detailed structure and functions of the various tables and modules that make up the security system software set will be described in part 2 of this 3 part series.■

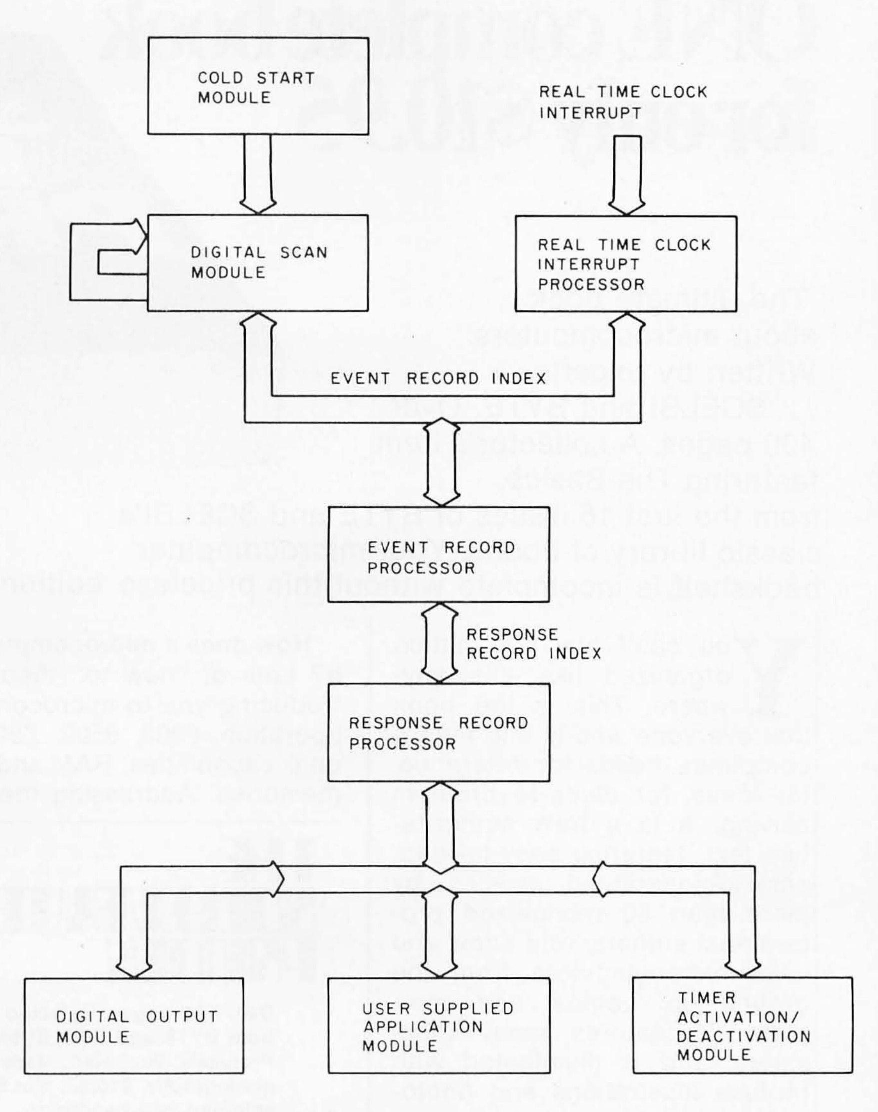


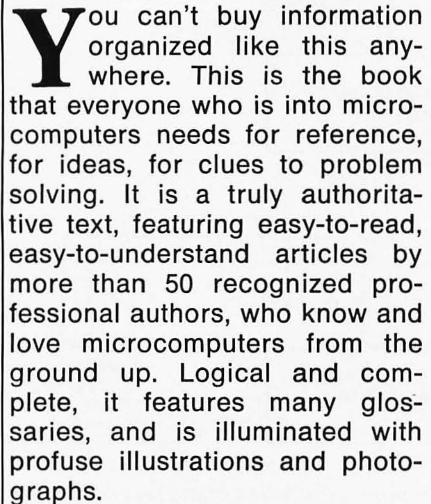
Figure 5: Block diagram of the computer controlled home security system.

Everything you've ever wanted to know about microcomputers in ONE complete book for only $10.95

Over 400 pages. Full 8½" x 11" size.

THE SCELBI BYTE PRIMER

The ultimate book about microcomputers. Written by experts ... SCELBI and BYTE. Over 400 pages. A collector's item, featuring The Basics from the first 16 issues of BYTE and SCELBI's classic library of books. Your microcomputer bookshelf is incomplete without this priceless edition.



You can't buy information organized like this anywhere. This is the book that everyone who is into microcomputers needs for reference, for ideas, for clues to problem solving. It is a truly authoritative text, featuring easy-to-read, easy-to-understand articles by more than 50 recognized professional authors, who know and love microcomputers from the ground up. Logical and complete, it features many glossaries, and is illuminated with profuse illustrations and photographs.

The Scelbi/BYTE Primer is divided into four logical sections, that take you from point "0" through building and programming your own computer... step-by-step-by-step.

What can you do with a microcomputer? Checkbook balancing. Recipe converting and food inventory. Heating and air conditioning control. Home and business security and management. Playing the ponies. Analysis of the stock market. Maintaining massive data banks. Self-instruction. Toys and games. Small business accounting and inventory. And lots, lots more.

How does a microcomputer do it? Lots of "how to" theory. Introducing you to microcomputer operation. 6800, 6502, Z80 CPU chip capabilities. RAM and ROM memories. Addressing methods.

THE SCELBI/BYTE PRIMER

Over 400 pages. Selected articles from BYTE and SCELBI books. Profusely illustrated. Many photographs. $10.95, plus $1 shipping and handling.

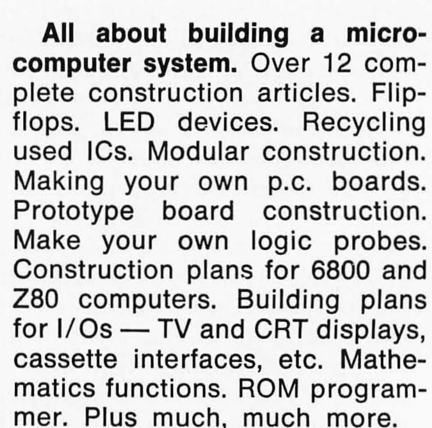
How to control peripherals. Transmission of information to and from computers. Magnetic recording devices for bulk storage. Analog to digital conversion. How a computer can talk. Other I/O techniques. And more.

Order your copy today!

SCELBI COMPUTER CONSULTING INC.
Post Office Box 133 PP STN
Dept. B
Milford, CT 06460

BITS
25 Route 101 West
P.O. Box 428
Peterborough, NH 03458
1-800-258-5477

Prices shown for North American customers. Master Charge. Postal and bank Money Orders preferred. Personal checks delay shipping up to 4 weeks. Pricing, specifications, availability subject to change without notice.



All about building a microcomputer system. Over 12 complete construction articles. Flip-flops. LED devices. Recycling used ICs. Modular construction. Making your own p.c. boards. Prototype board construction. Make your own logic probes. Construction plans for 6800 and Z80 computers. Building plans for I/Os — TV and CRT displays, cassette interfaces, etc. Mathematics functions. ROM programmer. Plus much, much more.

How to program a microcomputer. Programming for the beginner. Assembling programs by hand. Monitoring programs. Number conversions. Game of Hexpawn. Design your own assembler. Lots more.

And that's only the beginning! Others have spent millions acquiring the type of microcomputer information found within the 400 pages of **The Scelbi/BYTE Primer.** But, it costs you only $10.95, plus $1 for postage and handling, complete! You know the quality of Scelbi and BYTE. This is your assurance of excellence throughout this MUST text. Order your copy today! And, get one for a friend!

First Time Offer for the Micro Market





MODEL 501

DATA-SCREEN® TERMINAL

A NEW LOW COST, MICROPROCESSOR CONTROLLED CRT TERMINAL

QUALITY — APPEARANCE ECONOMY

ONLY $995.⁰⁰ (QTY 1)
DELIVERED CONT. USA

90 DAY WARRANTY GUARANTEES YOU YOUR EQUIPMENT WORKS PROPERLY

1 YEAR EXTENDED WARRANTY AT SPECIAL RATES



✓ Our Specifications Before You Buy

SPECIFICATIONS

DISPLAY

SCREEN CAPACITY, CHARACTERS	2000
CHARACTERS PER LINE	80
NUMBER OF LINES	25
SCREEN	P4 phosphor (white)
TUBE SIZE(DIAGONAL)	12 inches (30.4 cm)
VIEWING AREA	54 square inches (137.1 cm)
CHARACTER SIZE	0.20" high x .08" wide (5.08 mm high x 2.03 mm wide)
REFRESH RATE	60 Hz (50 Hz available)
SCAN METHOD	Raster
CHARACTER GENERATION	5 x 7 character in an 8 x 10 dot matrix
CURSOR	Blinking block

MEMORY

TYPE	Random Access Memory
CAPACITY	2000 characters

OPERATOR CONTROLS

POWER ON/OFF SWITCH	On rear of unit
BRIGHTNESS CONTROL	On rear of unit

POWER REQUIREMENTS

Model 501 — 115 volts, 60 Hz, 100 watts nominal
Model 502 — 230 volts, 50 Hz, 100 watts nominal

INTERFACE

DATA FORMAT	
DATA BITS	7 serial, asynchronous
DATA BIT 8	1, 0 or deleted
PARITY	Odd, even or deleted with error displayed as DLE
STOP BITS	1 or 2
DATA TRANSFER RATE	50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600 BAUD

STANDARD FEATURES

INVERSE VIDEO	Operator or software selectable
TRANSMIT MODES	Half or full duplex (switch selectable)
DATA ENTRY	Top or bottom line
END OF LINE BELL	Switch selectable
CURSOR POSITIONING	X–Y
CURSOR ADDRESS	Load and read
DISPLAYABLE CHARACTERS	126 (including space)
CURSOR CONTROLS	Up, down, left, right, home, return
AUTOMATIC ROLL–UP	Switch selectable
AUTO CARRIAGE RETURN AND LINE FEED	Switch selectable
MONITOR MODE	Special "Monitor" Mode allows display of control codes (first two columns) of ASCII Code Chart).

NATIONAL DISTRIBUTOR FOR



DEALER NETWORK IS NOW BEING SET UP — YOUR INQUIRY IS INVITED





TEC®

OLDEST INDEPENDENT COMMERCIAL CRT TERMINAL MANUFACTURER

3S SALES, INC. P.O. Box 45944 Tulsa, OK 74145

918 - 622-1058

Circle 379 on inquiry card.

An Exposure to MUMPS

David D Sherertz
University of California
San Francisco CA 94143

About the Author

David D Sherertz is a senior systems analyst in medical information science at the University of California, San Francisco. His activities include design analysis, medical programming and technical writing. For the past three years he has been actively involved in the MUMPS standardization effort, and is currently chairman of the Subcommittee for Language Development within the MUMPS Development Committee.

MUMPS (Massacusetts General Hospital Utility MultiProgramming System) was developed in the mid 1960s as a timesharing system for a modest minicomputer. A general purpose interpretive language, also called MUMPS, is an integral part of the system. The MUMPS language contains a powerful set of string manipulation capabilities, a pattern matching facility and timing functions, all of which facilitate the creation of conversational programs. In addition, users can share access to a hierarchically organized data base, with the file management details transparent to the user.

Recently, the American National Standards Institute approved the MUMPS programming language as a national standard, joining FORTRAN, COBOL, and PL/I in that category. The existence of a MUMPS standard and the robustness of the features included in the standard language make MUMPS an attractive choice for the development of interactive applications. This article gives an overview of Standard MUMPS, emphasizing its capabilities.

Commands and Variables

The MUMPS language is based on the set of commands summarized in table 1. Any command name can be abbreviated to its first letter (the HANG command is distinguished from HALT by the presence of an argument). Supplemental commands are usually added by implementors to support program creation, editing, debugging and storage. The execution of any command (except IF, ELSE or FOR) can be made conditional by adding a logical expression to the command word. The conditional command is then executed only if this "postconditional" is true. A maximum time limit can be placed on the actions of certain commands (LOCK, OPEN and READ). For example, the command OPEN TAPE:30 will periodically attempt to acquire the device TAPE for 30 seconds. If the unit is not available within this time, the command proceeds as if it were completed and a special variable is set by the system to allow checking for the expiration of the timer.

Command	Provides for
BREAK	access points for programming and debugging aids
CLOSE	relinquishing device ownership
DO	generalized subroutine calls
ELSE	conditional execution
FOR	generalized iterative execution
GOTO	generalized transfer of control
HALT	terminating execution
HANG	suspending execution for a specified period of time
IF	conditional execution
KILL	expunging specified variables and their values
LOCK	a generalized interlock facility for concurrent processes
OPEN	securing device ownership
QUIT	explicit termination of a DO or FOR command
READ	specifying data input
SET	assigning values to variables
USE	designating a specific device for input and output
VIEW	an access point for examining machine-dependent information
WRITE	specifying data output
XECUTE	execution of string expressions as program text
Z...	implementation-specific extensions

Table 1: Commands in the MUMPS language. The execution of any command can be made conditional by adding a logical expression to the command word. MUMPS is a general purpose interpretive language developed at Massachusetts General Hospital and designed for use in writing conversational programs. It features string manipulation, pattern matching ability and timing functions.

INTRODUCING

DUAL DRIVE MINIFLOPPY FOR PET!

A FULL RANGE OF PERIPHERALS TO COMPLETELY SUPPORT YOUR PET + PROFESSIONAL SOFTWARE SUPPORT = A TOTALLY INTEGRATED PET* SYSTEM



DOMESTIC & OVERSEAS DEALER INQUIRIES INVITED ON DISK SYSTEM & MEMORY

NOW THE DISKDRIVER DUAL DRIVE DISK SYSTEM CAN ADD TOTAL SYSTEM CAPABILITIES TO PET!

- ★ DUAL MINI FLOPPY DRIVE (5.5") WITH 100K PER DISK SIDE FOR TOTAL 200K ON LINE.
- ★ EXPANDABLE IN MARCH/APRIL 1979 TO 400K ON LINE VIA DOUBLE DENSITY FORMAT.
- ★ DISKMON-DISK OPERATING SYSTEM (DOS) IS RESIDENT IN PROM VIA DISK CONTROLLER BOARD THAT PLUGS DIRECTLY INTO THE INTERNAL EXPANDAPET MEMORY BOARD.*
- ★ DISKMON COMMANDS EXIST SIMULTANEOUSLY WITH THE BASIC LANGUAGE COMMANDS.
- ★ DISKMON ADDS 16 COMMANDS TO BASIC INCLUDING DISK DATA FILE COMMANDS.
- ★ DESIGNED FOR COMMERCIAL SPEED REQUIREMENTS — READS AN ENTIRE TRACK AT A TIME.
- ★ DISK DRIVES ARE AUTOMATICALLY TURNED OFF BETWEEN COMMANDS FOR LESS WEAR.
- ★ EXCELLENT DOCUMENTATION AND FULL SOFTWARE SUPPORT VIA FUTURE SOFTWARE.
- ★ DISKMON AUTOMATICALLY SUPPORTS COMMERCIAL TYPE PRINTER (CENTRONICS MODELS) VIA PARALLEL PORT.
- ★ DISKDRIVER COMES COMPLETE WITH ATTRACTIVE BEIGE CABINET, BUILT IN POWER SUPPLY, DISK CONTROLLER BOARD, ALL PLUG COMPATIBLE WITH YOUR PET (DISKDRIVER REQUIRES THE EXPANDAPET)
- ★ NOW FOR THE FIRST TIME YOUR PET CAN BE UTILIZED AS A COMPLETE, TOTAL SYSTEM.
- ★ APPLICATION/DEVELOPMENT SOFTWARE AVAILABLE ★ BUSINESS PACKAGES IN MARCH/APRIL.
- ★ FULL PRODUCT (HARDWARE & SOFTWARE) SUPPORT FROM COMPUTHINK AND DEALERS.

THE DISKDRIVER, DOS, AND EXPANDAPET ARE ALL PRODUCTS OF COMPUTHINK

HARDWARE/SOFTWARE PRODUCT DIRECTORY

DKH642-1	DISKDRIVER DUAL DRIVE MINIFLOPPY SYSTEM (200K ON LINE STORAGE)	$1295
EXPMEM	EXPANDAPET MEMORY EXPANSION BOARD (SPECIFY 16K, 24K, 32K) 16K MODEL	$ 425
PRT200	CENTRONICS COMMERCIAL PRINTER (MODEL 779-1) WITH FRICTION FEED	$1245
PRT202	CENTRONICS COMMERCIAL PRINTER (MODEL 779-2) WITH TRACTOR FEED	$1345
PRT100	AXIOM HOBBYIST PRINTER (ELECTROSTATIC 5.5" WIDE PAPER) APPROX.	$ 450
NPK101	NEECO PERIPHERAL KEYBOARD (TYPEWRITER TYPE) AVAILABLE JAN.	$ 140
ASG200	PET ASSEMBLER PROGRAMMER'S GUIDE (SHOWS HOW TO TAP INTO BASIC)	$19.95
ASM789D	PET ASSEMBLER 6502 PROGRAMMING SOFTWARE (ON DISK—WITH MANUAL)	$49.95
LNK456D	AUTOLINK PROFESSIONAL LINKING LOADER SOFTWARE (ON DISK—WITH MANUAL)	$49.95
EDT392D	ASSEMBLER-EDITOR, ASSEMBLER SOURCE PROGRAM EDITOR (ON DISK—WITH MANUAL)	$49.95
DUG078	DISKMON USER'S GUIDE (PROVIDED WITH SYSTEM—CAN BE PURCHASED SEPARATELY)	$ 9.95
DKL067	DISKMON (DOS) ASSEMBLER LISTING IN MANUAL FORM	$19.95
FOR112	FORTRAN COMPILER FOR DISKDRIVER SYSTEM (AVAILABLE JAN./FEB.)	$69.95
PLM118	PROFESSIONAL PLM COMPILER FOR DISKDRIVER (AVAILABLE JAN./FEB.)	$49.95
FUTURE	BUSINESS AND PROFESSIONAL PACKAGES STARTING IN JAN./FEB.	

* **NOTE** PET IS A TRADEMARK OF COMMODORE AND IS SOLD ONLY BY AUTHORIZED DEALERS.

NEECO

EASTERN U.S. & EUROPEAN
NEW ENGLAND ELECTRONICS
248 BRIDGE ST.
SPRINGFIELD, MASS., 01103
(413) 739-9626

COMPUTHINK

WESTERN U.S. & CANADA
COMPUTHINK
3260 ALPINE ROAD
MENLO PARK, CA., 94025

Circle 281 on inquiry card.

GLOSSARY

Argument: The independent variable upon whose value a function depends. For the function y=cos(x), x is the argument of the function.

Binary Operator: An operator which acts on two operands. For example, the addition operator (+) is a binary operator. Also known as a dyadic operator. Compare *Unary Operator*.

Binding: The action of associating machine addresses with the symbolic addresses of a computer program.

Case Statement: A type of "multiple choice" statement in which the execution of the statement depends on the numerical value of an argument contained in the statement.

Global Variable: In MUMPS, a variable that can be defined by any authorized user and can be used (referenced) by any user's program.

Literal: A constant value (as opposed to a variable).

Local Variable: In MUMPS, a variable that can be defined and used by only one user and that user's program (which may consist of many segments).

Scope: The portion of a program in which the definition of a variable (but not necessarily its value) remains unchanged.

Truncate: Usually means to reduce the precision of a number by arbitrarily removing low or high order digits without adjusting the remaining digits. The number 4.5673 could be truncated to 4.5, for instance.

Variable: A name or symbol which in principle can take an arbitrary value during execution of a program.

MUMPS contains only one explicit data type: the variable length character string. Certain contexts require an implicit interpretation of a value as a real number, an integer or a logical value. Unambiguous rules are defined for mapping a string into the requisite data type. Variables in MUMPS are created dynamically by assigning values to them (as in APL and BASIC). There is no scoping or binding of variables. Variables exist across program boundaries and can only be removed with the KILL command.

Each MUMPS user has a private variable space, called "local variables" which only that user can access and modify. There is also a common data base called "global variables" which any authorized user can inspect or change. Both local and global variables can be subscripted to any depth; subscripted variables are stored as sparse arrays, with storage allocated only for defined subscripts. A third class of variable is called the "special variables." These are reserved variables, distinguished by a leading dollar sign ($). The special variables are maintained by the MUMPS system for a variety of purposes, including line and column control on IO devices, current data and time, process identification and timers.

Date Base Management

MUMPS global variables, or globals, are tree structured files in which any node may contain a data value. Syntactically, globals are distinguished from local variables by a circumflex (^) preceding the global name. Operationally, globals form a hierarchical data base that exists independently of any user. References to a specific global node appear as a list of subscripts. For example, the sales record of a salesman might be accessed with ^SALESMEN (DEPT,LNAME, FNAME), where DEPT, LNAME and FNAME have appropriate values. Each subscript level in a global reference represents a lower node in the tree structure. There is also a facility in MUMPS, called the *naked reference*, to avoid redundant higher level subscripts in global references.

When a new node is created by assigning a value to it, the necessary access paths to the node are constructed automatically by the system. Similarly, when an individual node or an entire subtree of a global is expunged with the KILL command, the affected paths are pruned appropriately by the system. Thus, the MUMPS file manager achieves a substantial degree of data independence, allowing a programmer to concentrate on building a logically meaningful file structure. The details of the physical file layout are invisible to the programmer.

Operators and Pattern Matching

The MUMPS language includes the normal set of arithmetic unary operators (+ and −) and binary operators (+ − X and /). There is also a modulo operator (#), a division with truncation operator (\), the algebraic relational operators less than (<), equals (=) and greater than (>), as well as the logical operators and (&), or (!) and not (').

The language includes several string operators which greatly simplify string manipulation. Strings can be merged with the concatenation operator (_), and compared with the relational operators equals (=),

The Software Company

Technical Systems Consultants, Inc.

TSC, Technical Systems Consultants, is **the software company** for all the newest, most innovative ideas in computer software. TSC builds a variety of programs, packages and games so you can get down to business or just some fun.

Text Editing System

The most complete and versatile editor available for the 6800 and 8080 micro. The system is line and content oriented for speed and efficiency and features such commands as block move and copy, append and overlay, as well as string manipulators. The 6800 version requires 5K beginning at 0 hex, the 8080 needs 6K starting at 1000 hex. Both should have additional file space as required.

SL68-24	**6800 Text Editing System**	**$23.50**
SL68-24C	**w/cassette**	**$30.45**
SL68-24P	**w/paper tape**	**$31.50**
SL68-24D	**w/mini flex disc**	**$31.50**
SL68-24F	**w/flex disc**	**$50.00**
SL80-10	**8080 Text Editing System**	**$28.50**
SL80-10P	**w/paper tape**	**$37.50**
SL80-10F	**w/ CP/M disc**	**$40.00**

Space Voyage

SL68-5C	**w/cassette**	**$18.95**
SL68-5P	**w/paper tape**	**$19.00**
SL80-9P	**w/paper tape**	**$19.00**

6800 Disassembler

SL68-27C	**w/cassette**	**$15.95**
SL68-27P	**w/paper tape**	**$13.00**

Micro BASIC Plus for 6800

SL68-19C	**w/cassette**	**$22.90**
SL68-19P	**w/paper tape**	**$21.95**

6800 Floating Point Package

SL-68-4	**Floating Point Package**	**$6.50**
SL68-4P	**w/paper tape**	**$9.50**

Text Processing System

As a complement to the Editor, the Processor supports over 50 commands for left, right or center justification, titling, paging and general text output formatting. A loop command is available for repeated formatting jobs such as form letters. Also included are capabilities for macro definition to build special formatting commands. The program requires about 8K of RAM and previously edited text.

SL68-29	**6800 Text Processor**	**$32.00**
SL68-29C	**w/cassette**	**$38.95**
SL68-29P	**w/paper tape**	**$40.00**
SL68-29D	**w/mini flex disc**	**$40.00**
SL68-29F	**w/flex disc**	**$75.00**
SL80-11	**8080 Text Processor**	**$32.00**
SL80-11P	**w/paper tape**	**$41.00**
SL80-11F	**w/ CP/M disc**	**$50.00**

Relocator

This self-prompting, easy to use program relocates object code in RAM or from tape. Complete instructions included for making the TSC Editor and Assembler or Editor and Text Processor co-resident. (As sold they reside in the same area.) Just over 1K in length.

SL68-28	**6800 Relocator**	**$ 8.00**
SL68-28C	**w/cassette**	**$14.95**
SL80-13	**8080 Relocator**	**$ 8.00**
SL80-13P	**w/paper tape**	**$13.00**

The Mnemonic Assembler

The ideal addition to the Text Editing System, together they form a complete program development center. The Assembler is one of the most versatile available and allows for easy adaptation to most systems. The Assembler is many times faster than other resident assemblers. Requires approximately 5.5K plus file and symbol table space.

SL68-26	**6800 Mnemonic Assembler**	**$23.50**
SL68-26C	**w/cassette**	**$30.45**
SL68-26P	**w/paper tape**	**$31.50**
SL68-26D	**w/mini flex disc**	**$31.50**
SL68-26F	**w/flex disc**	**$50.00**
SL80-12	**8080 Mnemonic Assembler**	**$25.00**
SL80-12P	**w/paper tape**	**$34.00**
SL80-12F	**w/ CP/M disc**	**$40.00**

All programs include complete source listing. Cassettes are in the Kansas City standard format.

All orders should include check or money order. Add 3% for postage and for orders under $10, please add $1 for handling. Send 25¢ for a complete software catalog.





Technical Systems Consultants, Inc.
Box 2574 W. Lafayette, IN 47906

Specialists in Software & Hardware for Industry & the Hobbyist

Circle 370 on inquiry card.

Function	Returns
$ASCII	as an integer the ASCII code of a selected character from a string
$CHAR	a string whose characters are generated from a specified set of integer ASCII codes
$DATA	an integer code that indicates if a specified local or global *variable* has a defined value or node path
$EXTRACT	a character or substring, extracted from a starting and ending character position in a string expression
$FIND	an integer that gives the end character position of a specified substring within a string expression
$JUSTIFY	a formatted string consisting of a right justified expression in a blank field of a specified size
$LENGTH	an integer which is the length in characters of a specified string expression
$NEXT	an integer which is the next lowest defined numeric subscript on the same level as the last subscript of a named subscripted variable
$PIECE	a string extracted from the characters between two specified occurrences of some substring delimiter within a string expression
$RANDOM	a pseudorandom integer from the set of integers within a specified interval range
$SELECT	the value of a particular expression from a set of expressions, selected by the first true expression in an associated set of logical expressions
$TEXT	a string that consists of the commands of a specified program line in the current routine
$VIEW	a value which is machine dependent data, accessed in an implementation-specific manner
$Z . . .	values from functions that are implementation-specific extensions

Table 2: Built-in functions included in ANSI Standard MUMPS, which can be nested to any arbitrary level.

contains ([),follows (]). (As an explanation, assuming A and B are defined variables, then the expression A[B is true if and only if B is a substring of A: similarily, A]B is true if and only if A follows B in the ASCII collating sequence.)

The logical operator not (') can be used alone or in conjunction with the relational and logical operators above to produce the complementary operation. For example, A'>B means A less than or equal to B, and A']B means A does not follow B. All logical and relational operators produce an integer "truth value" (1 is true, 0 is false).

A particularly powerful feature in MUMPS is the pattern matching facility. A pattern specification follows the operator for pattern matching, which is a question mark (?). The pattern matching operator determines whether or not the string preceding it conforms to the specified pattern. Patterns are formed by combining any number of pattern codes or string literals. A string literal is any sequence of graphics enclosed in quotes (" "). Each element of a pattern is preceded by an integer repetition count. An indefinite count is specified by a period (.). Included in the seven pattern codes are A, which matches alphabetics, N, which matches numerics, and P, which matches all other graphics. The pattern matching facility allows filters to be constructed, similar to those of the language SNOBOL, with which to test input for validity. The degree of selectivity of the filter is controlled by the amount of specificity in the pattern. For example, the pattern

2N1P3A.A1P2N.N

means that an acceptable string must contain two numbers followed by one graphics symbol, three alphabetic characters, any number of alphabetic characters, one graphics symbol, two numbers, and finally any number of numbers. The following strings would pass successfully through this filter:

"25 AUG 1948"
"07-MAY 1977"
"04/FEBRUARY/47"

On the other hand, the string filter 2N1"DEC 19"2N would only pass strings of the form "nnDEC 19nn".

Functions and Expressions

Table 2 summarizes the built-in functions included in Standard MUMPS. Because these functions can be nested to an arbitrary level, very complex string manipulations can be performed in a single statement.

Expressions in MUMPS are made up of literals, functions, and variables, combined with operators that act upon them.

TRS-80 OWNERS

AVAILABLE FOR IMMEDIATE DELIVERY

CASSETTE SOFTWARE DISKETTE

Package # 1036 (Level II) $495.00

COMPLETE SMALL BUSINESS – This program is a complete small business program that was tailored to work for most small business applications. The program includes such things as Accounts Receivable, Accounts Payable, Invoicing, Inventory Control, Payroll and General Ledger.

Package # 1038 (Level II) . . . . $ 99.95
ACCOUNTS RECEIVABLE

Package # 1039 (Level II) . . . . $ 99.95
ACCOUNTS PAYABLE

Package # 1044 (Level II) $125.00
INVENTORY CONTROL

Package # 1045 (Level II) . . . . $ 99.95
INVOICING

Package # 1046 (Level II) . . . . $ 99.95
PAYROLL

Package # 1047 (Level II) . . . . $ 99.95
MAILING LIST

Package # 1024
(Level II, DISKETTE) $24.95
Includes the following:
SPACEWAR I – BANNER – UFO ATTACK – PILE UP – BIORHYTHM – AUTO RACE and WORDS.

Package # 1026
(Level II, DISKETTE) $24.95
Includes the following:
SPACEWAR II – CIVIL WAR – TRAP THE TRIBBLE – LIFE – KNIGHT – CONCENTRATION and LUNAR LANDER.

OVER 150 EXCITING PROGRAMS
★ MORE EVERY DAY ★

EDUCATION ELECTRONICS MARKETING

Also available for PET and APPLE.
All SOFTWARE-80 programs are guaranteed.
Programs available in Level I and Level II Basic.
All of our business programs will operate with printer; custom programs also available.
With certified check or money order, all orders shipped within 24 hours.
Personal checks allow 2 weeks. Send for our complete catalogue.

SOFTWARE-80

18228 Cabrillo Court
Fountain Valley, CA 92708
(714) 962-3423

ALL PRICES AND PROGRAMS ARE SUBJECT TO CHANGE WITHOUT NOTICE

Circle 321 on inquiry card.

Since the only data type is a string, any operator can be used on any operand. The interpretation of an operand is automatically determined by the implicit data type required by its operator. All expressions are evaluated in strict left to right order; there is no precedence of operators. Parentheses can be used to override this convention.

Other Features

Another feature included in MUMPS allows data to be treated as program code: indirection, although usually found only in assembly languages as a method of address calculation, is available in MUMPS in a powerful symbolic form. In effect it allows implementation of the functions of CASE statements, subroutine parameter passing, and a number of other useful language features that are not *explicitly* included in MUMPS. Both variable names and command arguments can be used indirectly by putting an "at" sign (@) before the element. This forces the intrepreter to use the value of the variable to search for another variable. For example, if the variable AMOUNT3 has the value 45.82, and the variable SWITCH has the value "AMOUNT3," then the command WRITE @SWITCH would write 45.82.

An additional form of indirection is provided by the XECUTE command, which permits general expressions to be executed as program text. One common use of XECUTE is to direct a driver program, in which the next program step is determined dynamically by branching logic in prior steps, using lines of program code stored previously as data values in a global file. This approach has been utilized in constructing automated medical histories, in computer assisted instruction, and in text editors and other utility programs written in MUMPS.

The MUMPS Environment

Traditionally, MUMPS has been implemented on dedicated minicomputers with the primary memory divided into modest (2 to 6 K bytes) user partitions. The central processor is timeshared among the users, but since most MUMPS applications are usually IO bound, the processor can handle as many as 40 users and still maintain a rapid response time. Over the past few years, MUMPS has been implemented within existing general timesharing systems on both large and small computers.

Because the user partitions are relatively small, a programmer must explicitly segment large MUMPS applications into appropriately sized routines, using the GOTO command to overlay segments as required. Recent implementations have eased the size restriction on user partitions by using techniques such as sharing commonly used routines among several partitions, and pooling storage for the local variable values from all partitions.

Summary

MUMPS is a useful development tool for many applications involving string manipulation, interactive dialogues, and file management. Multi-user access to a hierarchical data base is a particularly powerful component of the language. The number of MUMPS implementations has been steadily growing, including several supported by major computer vendors. Although a majority of the present MUMPS applications are related to health care because of the language's historical origins, nothing constrains MUMPS solely to medical systems. In fact, an increasing number of business applications have been implemented in recent years using MUMPS.

The attractiveness of MUMPS is primarily due to the full set of facilities available for conversational applications. The collection of useful interactive capabilities included in the MUMPS Standard is much more complete than that found in any other interpretive language. Among the advantages of the features of Standard MUMPS are the following:

String Handling: MUMPS includes an extensive set of string operations and functions that makes string manipulations quick and easy. Tasks such as string searches, sorts, data extractions and compactions are greatly simplified by these facilities. Furthermore, MUMPS allows variable length strings to be used routinely in programs, without the obstables presented by most other programming languages.

Globals: MUMPS eliminates many of the problems of working with random access devices by obviating the need for traditional read and write operations as well as the intermediate abstraction of a file. A data element on a secondary storage medium is directly referenced as a set of subscripts, with all the details of file organization and retrieval handled by the system. Nothing with the flexibility of this unique hierarchical file system can be found in any other general purpose programming language. Languages such as FORTRAN or BASIC, the most obvious alternatives to MUMPS on small computers, allow for random file operations, but only as extensions that are often machine dependent.

Pattern Matching: MUMPS allows user

input to be filtered with a useful pattern matching facility. This helps to minimize the effect of user or device errors. The pattern matching capability is sufficiently general to be used in various transaction oriented applications, such as constructing command processors. It is even powerful enough for the development of other language scanners.
Timing: MUMPS enables a programmer to associate timing constraints with several operations. If a READ command does not get a complete response from a user terminal within a specified time after queuing the request, the command will take any input already received and proceed. This feature allows testing for terminal malfunctions as well as prompting users in a time critical dialogue. A similar capability is available for detecting and handling delays in accessing globals or acquiring shared resources as a result of multi-user interference.
Program Management: MUMPS provides all the facilities needed to manage programs and program files. Programs can be created, edited, cataloged and debugged from within MUMPS. As in APL, and similar programming environments, MUMPS incorporates the necessary tools to make a complete programming system. The task of debugging programs in MUMPS is made easier by facilities that allow the interruption of executing programs, the examination or modification of variables or program text, and the resumption of program execution at a different location.
Economy: MUMPS implementations are available on a variety of small, relatively inexpensive minicomputers and microcomputers. These dedicated MUMPS computers can be managed with a small amount of administrative overhead. There are numerous commercially available BASIC systems roughly comparable to these MUMPS installations. However, for many applications MUMPS represents a more economical and advanced tool for system development.
Applications Experience: The number of MUMPS application packages is steadily increasing, with many developed and available in the public domain. There is also an active MUMPS Users' Group (MUG) in the United States as well as in Europe and in Japan. MUG publishes newsletters, holds an annual meeting, distributes MUMPS documents and maintains a library of available MUMPS application programs. The more successful and widespread MUMPS appplications include an abbreviated medical record system for outpatient clinics, numerous clinical laboratory reporting systems, and several inventory and information management systems.

In conclusion, MUMPS is designed for an interactive timeshared environment, and for a general class of conversational applications. On the whole, MUMPS satisfies the requirements of these areas more completely than many other languages.■

A full specification of the MUMPS Language Standard is available from the MUMPS Users' Group (the cost is $5.00). Write for ANSI X11.1-1977 to:

MUMPS Users' Group
D-130
POB 208
Bedford MA 01730

Information on MUG and MUG documents can be obtained from:

Richard E Zapolin
MUG Executive Director
MUMPS Users' Group
D-130
POB 208
Bedford MA 01730

Technical questions on the MUMPS Standard should be directed to:

Dr Richard F Walters
Chairman, MUMPS Development Committee
Medical Learning Resources
University of California at Davis
Davis CA 95616

Thomas E Doyle
5222 Big Bow Rd
Madison WI 53711

A Computerized Mailing List

Creating, printing and updating mailing lists offer practical applications for a microcomputer system with a floppy disk or other magnetic mass storage medium. The series of programs presented here allow the user to create and maintain mailing lists. The programs are written to be run on a 8080 or Z-80 system with at least one disk drive and 20 K of programmable memory. The program is written for an interpreter called BASIC-E which is widely distributed under several names.

One of the biggest problems facing a programmer using BASIC-E is the relative lack of complete documentation on the file commands. A fairly detailed description of the BASIC-E file commands, used in the programs, will be included to help this situation.

There are six basic types of operations necessary to maintain mailing lists:

1. Generate a new list. This operation would normally be performed once for each mailing list.
2. Print the list. This operation would be used as copies of the list are needed, or as changes to the list are made.
3. Add to the list. New entries to the list would be added to the end of the list with this operation.
4. Update existing entries. Changes could be made to entries already in the list with this operation.
5. Insert new entries into list. This operation would allow new entries to be inserted into the list. All existing entries would be shifted to make room for the new entry.
6. Remove an entry from the list. An existing entry could be removed from the list and the remaining entries on the list would be moved up to fill in the space.

Seven programs were written to perform these functions (see listings 1 thru 7). Two of the programs print the list in differrent formats.

Sections

Each entry (or record) in the mailing list consists of seven sections:

1. Name
2. Call sign
3. Street address
4. City
5. State
6. Zip code
7. Phone number

These programs were written to handle the mailing list for an amateur radio club, so the ham radio call sign section was included. This section could easily be removed or additional sections added, as needed, in a specific application. Keeping the entries divided into sections allows maximum flexibility in the system.

The mailing list is set up as a disk file divided or "blocked" into records. Each record occupies exactly the same amount of space on the disk. Files set up in this way are not the most efficient in terms of using the space available on the disk, but this type of file allows direct access to any

Circle 216 on inquiry card.

record in the file using a record sequence number as an address. Files of this type are also referred to as random access files.

128 bytes are set up for each entry in the file. If the entry does not use all 128 bytes, the remaining bytes are filled with blanks.

Listing 1: Program used to generate a new mailing list. The user is prompted to enter the necessary information. (This program and the other programs in this article were written in BASIC-E.)

```
PRINT "PROGRAM #1 - GENERATE NEW LIST -"

     INPUT "FILE NAME";FILE.NAME$
     FILE FILE.NAME$(128)

FOR RECORD=1 TO 1000
     INPUT "NAME";N$
     IF N$="END" THEN 100
     INPUT "CALL";C$
     INPUT "STREET";S$
     INPUT "CITY";T$
     INPUT "STATE";A$
     INPUT "ZIP";Z$
     INPUT "PHONE";P$
     PRINT #1,RECORD;N$,C$,S$,T$,A$,Z$,P$
NEXT RECORD

100 END
```

Listing 2: A program which prints the mailing list, single column.

```
PRINT "PROGRAM #2 - TYPE LIST, SINGLE COLUMN -"

     INPUT "FILE NAME";FILE.NAME$
     FILE FILE.NAME$(128)
     IF END #1 THEN 200

     INPUT "PRINT PHONE NUMBERS ";PHONE$
     PRINT:PRINT:PRINT

FOR RECORD=1 TO 1000 STEP 1
     READ #1,RECORD;N$,C$,S$,T$,A$,Z$,P$
     PRINT C$;TAB(9);N$
     PRINT S$
     PRINT T$;", ";A$;"  ";Z$
     IF PHONE$="YES" THEN PRINT P$
     PRINT
NEXT RECORD

200 END
```

Listing 3: A program which prints the mailing list, double column.

```
PRINT "PROGRAM #3 - PRINT LIST, DOUBLE COLUMN -"

     INPUT "FILE NAME";FILE.NAME$
     FILE FILE.NAME$(128)
     IF END #1 THEN 200

     INPUT "PRINT PHONE NUMBERS";PHONE$

     PRINT:PRINT:PRINT

FOR RECORD=1 TO 1000
     FLAG=0
     READ #1,RECORD;N$,C$,S$,T$,A$,Z$,P$
     FLAG=1
     READ #1,RECORD+1;NE$,CE$,SE$,TE$,AE$,ZE$,PE$
     PRINT C$;TAB(9);N$;TAB(34);CE$;TAB(43);NE$
     PRINT S$;TAB(34);SE$
     PRINT T$;", ";A$;"  ";Z$;TAB(34);TE$;", ";AE$;"  ";ZE$
     IF PHONE$="YES" THEN PRINT P$;TAB(34);PE$
     RECORD=RECORD+1
     PRINT:PRINT
NEXT RECORD

200 IF FLAG=0 THEN 300
    PRINT C$;TAB(9);N$
    PRINT S$
    PRINT T$;", ";A$;"  ";Z$
    IF PHONE$="YES" THEN PRINT P$
300 PRINT:PRINT:PRINT:PRINT
    END
```

The first step in writing a BASIC-E program that uses disk files is to set up, or open, the file. The FILE statement is used to open a file. It is used in the following form to set up a blocked file: FILE NAME$ (RECORD LENGTH). Each of the seven programs starts with the following lines to set up the file:

```
INPUT "FILE NAME"; FILE.NAME$
FILE FILE.NAME$ (128)
```

The first line enables the user to type in the name of the file containing the mailing list. The file name is stored as string variable FILE.NAME$. The second line opens the file as a blocked file with a record length of 128. BASIC-E searches through the files on the disk looking for a file with the name typed in by the user. If a file with a matching name is found, it is used. If no file with a matching name is found, BASIC-E creates a file with the name typed in by the user. From this point on, references to the file are made by referring to the file as #1, rather than by name.

The IF END statement is an important statement in BASIC-E programs that read files from a disk. It is of the form IF END #N THEN XXXX, where N is the file number and XXXX is the line number the program should jump to if the end of a file is encountered. If this statement is not included in the program and the end of a file is encountered during execution, the program terminates and prints an error message.

Data is written into the disk file using the PRINT statement. It is of the form PRINT # N, RECORD; V1, V2, V3, . . ., VN, where N is the file number, record is the record number, and V1 thru VN is the data to be written into the file. The variables can be numeric values or strings. All data for the mailing lists is stored as strings.

Data is read from the disk file with the read statement. The statement is of the form READ # N, RECORD; V1, . . ., VN where N is the file number, record is the record number, and V1 thru VN are the variables.

The FILE, IF END, PRINT # N AND READ # N commands are the only ones used in the program that refer specifically to the disk file. BASIC-E also includes the CLOSE command, used to close a file. This command is not needed in these programs, since BASIC-E closes all files when the program ends.

Program 1

The first program, program 1 (see listing 1), is used to set up a new mailing list. The first two lines set up the file as a block-

ed file with a record length of 128. The data for name, call sign, street, city, state, zip code and phone number is entered. After the phone number is entered, all the information for that entry is stored on the disk. Additional sequential entries are made until the word END is entered for a name. Additional characteristics for each entry could be included by adding additional input statements and adding the variable to the PRINT #1, . . . statement. The order of the variables in the PRINT #1 statement is very important. The data is stored in the record in the order that the variables are listed in the statement. If a particular piece of information is not available for an entry, a dash or some other character should be entered to keep the order of the data in the record constant.

Listing 4: A program used to add new names, addresses, etc, to the list.

```
PRINT "PROGRAM # 4 - ADD TO LIST -"

     INPUT "FILE NAME"; FILE.NAME$
     FILE FILE.NAME$(128)
     IF END #1 THEN 100

REM FIND END OF FILE

  FOR RECORD=1 TO 1000
     READ #1,RECORD;N$
  NEXT RECORD

100 REM ADD TO END OF FILE

  FOR RECORD=RECORD TO 1000
     PRINT:PRINT
     PRINT "NEXT RECORD ";RECORD
     INPUT "NAME";N$
     IF N$="END" THEN 200
     INPUT "CALL";C$
     INPUT "STREET";S$
     INPUT "CITY";T$
     INPUT "STATE";A$
     INPUT "ZIP";Z$
     INPUT "PHONE";P$
     PRINT #1,RECORD;N$,C$,S$,T$,A$,Z$,P$
  NEXT RECORD
200 END
```

Listing 5: A program used to update the mailing list.

```
PRINT "PROGRAM #5 - UPDATE LIST -"

     INPUT "FILE NAME";FILE.NAME$
     FILE FILE.NAME$(128)
     IF END #1 THEN 200

     INPUT "CALL SIGN";CALL$

REM SEARCH FOR RECORD TO BE CHANGED

  FOR RECORD=1 TO 1000
     READ #1,RECORD;N$,C$
     IF C$=CALL$ THEN 100
  NEXT RECORD

100 PRINT "NEW INFORMATION."

     INPUT "NAME";N$
     INPUT "CALL SIGN";C$
     INPUT "STREET";S$
     INPUT "CITY";T$
     INPUT "STATE";A$
     INPUT "ZIP";Z$
     INPUT "PHONE";P$
     PRINT #1,RECORD;N$,C$,S$,T$,A$,Z$,P$

200 END
```

Program 2

The mailing list is printed by program 2 (see listing 2). The particular file containing the address information is selected by the first lines of the program. The user is then asked if phone numbers are to be included in the printout. Phone numbers would be printed if the list is being used for a membership list or directory. If the list is being used for mailing labels, phone numbers would most likely be omitted. Program 2 prints the contents of the file in the following format:

```
NAME 1              CALL 1
STREET 1
CITY 1              STATE 1 ZIP 1
PHONE 1 "if selected"

NAME 2              CALL 2
STREET 2
CITY 2,             STATE 2 ZIP 2
PHONE 2 "if selected"
```

Program 3

Program 3 (see listing 3) is used to print the list in the following format:

```
NAME 1              CALL 1        NAME 2              CALL 2
STREET 1                          STREET 2
CITY 1, STATE 1 ZIP 1             CITY 2, STATE 2 ZIP 2
PHONE 1 "if selected"             PHONE 2 "if selected"

NAME 3              CALL 3
STREET 3                          etc
CITY 3, STATE 3 ZIP 3
PHONE 3 "if selected"
```

This program is similar to program 2 except that the program reads two records from the disk at a time. The variable FLAG is used to insure that the program prints the last record if there is an odd number of records in the file.

Program 4

Additional records may be added to the end of the file with program 4 (see listing 4). After the file is selected and opened, records are read until the end of the file is located. After the end of the file is located, additional information is entered by the user and records are added sequentially to the end of the file. This operation continues until END is entered for a NAME.

Program 5

Changes in address, phone numbers or other sections of an existing record may have to be made. Program 5 (see listing 5) allows the user to select a particular record

Listing 6: A program used to insert new entries into the list.

```
PRINT "PROGRAM #6 - INSERT NEW ENTRY -"

     INPUT "FILE NAME"; FILE.NAME$
     FILE FILE.NAME$(128)
     IF END #1 THEN 100

     INPUT "CALL SIGN"; CALL$

REM FIND END OF FILE AND RECORD CONTAINING CALL

  FOR RECORD=1 TO 1000
     READ #1, RECORD; N$, C$
     IF CALL$=C$ THEN LOCATION=RECORD
  NEXT RECORD

     100 IF LOCATION=0 THEN PRINT "NOT FOUND": STOP
     FINISH=RECORD-1

REM MOVE RECORDS UP 1 STARTING AT LOCATION

  FOR RECORD=FINISH TO LOCATION STEP -1
     READ #1, RECORD; N$, C$, S$, T$, A$, Z$, P$
     PRINT #1, RECORD+1; N$, C$, S$, T$, A$, Z$, P$
  NEXT RECORD

REM INSERT NEW RECORD

  INPUT "NAME"; N$
  INPUT "CALL"; C$
  INPUT "STREET"; S$
  INPUT "CITY"; T$
  INPUT "STATE"; A$
  INPUT "ZIP"; Z$
  INPUT "PHONE"; P$

  PRINT #1, LOCATION; N$, C$, S$, T$, A$, Z$, P$

END
```

and reenter the information on that record. The user first enters the call sign which identifies the record to be changed. The program then reads through the file until a match is found. If no match is found the program ends. If a match is found the user may reenter the information for that record including any necessary changes.

Program 6

If the mailing list is kept in some logical order such as alphabetical, it may be necessary to insert records into the file. Program 6 (see listing 6) enables the user to insert a new record anywhere into the file. The user enters the call sign of the existing entry where the new record is to be inserted. The program then reads through the file looking for a match. When a match is found the record number is stored as variable LOCATION. The program continues reading through the file until the end of the file is located. All the records from LOCATION to the end of the file are moved up one record to allow room to insert the new record. The user enters the information for the new record which is then written into the file. This method of inserting new records into the file in their proper location

is much faster than adding the new records to the end of the file and running the file through a sorting routine.

Program 7

From time to time it may be necessary to eliminate records from the file. Program 7 (see listing 7) enables the user to locate and remove a record within the file by specifying a CALL SIGN. The program reads through the file looking for a record with a matching CALL SIGN. If a match is found, the record number is stored as variable REMOVE. The program continues reading through the file until the end of the file is located. ALL the records in the file from record number REMOVE to the end of the file are moved down one record, eliminating the gap that would have been left by the removal of the record. I hope this mailing list program will be of use to readers.■

Listing 7: A program used to remove entries from the list.

```
PRINT "PROGRAM #7 - REMOVE ENTRY FROM FILE -"

     INPUT "FILE NAME"; FILE.NAME$
     FILE FILE.NAME$(128)
     IF END #1 THEN 100

     INPUT "CALL SIGN";CALL$

REM SEARCH FOR RECORD AND END OF FILE

  FOR RECORD=1 TO 1000
     READ #1,RECORD;N$,C$
     IF CALL$=C$ THEN REMOVE= RECORD
  NEXT RECORD

     100 IF REMOVE=0 THEN PRINT "NOT FOUND" : STOP
         FINISH=RECORD-2

REM RE-PACK FILE

  FOR RECORD=REMOVE TO FINISH
     READ #1,RECORD+1;N$,C$,S$,T$,A$,Z$,P$
     PRINT #1,RECORD;N$,C$,S$,T$,A$,Z$,P$
  NEXT RECORD

REM BLANK OUT LAST RECORD
     PRINT #1,FINISH+1;" "," "," "," "," "," "," "

END
```

Additional Notes on BASIC-E

The following points will aid in converting programs written in Microsoft (MITS, Applesoft, OSI, Commodore) BASIC to BASIC-E.

BASIC-E does not allow IF-THEN statements to be located anywhere other than at the beginning of a line.

EXAMPLE: 100 A = 2.2: D = E + 1: IF A = D THEN 200

The above line would not be compiled by BASIC-E because the IF-THEN statement is not at the beginning of the line. The solution to this problem involves using the editor to insert a carriage return/line feed in place of the : preceeding the IF-THEN statement. After this is done the line will be in the following form which will be compiled correctly:

```
100 A = 2.2: D = E + 1
IF A = D THEN 200
```

BASIC-E uses the statement RND for random number generation. The editor can be used to replace all the RND (x) statements in the Microsoft BASIC program with RND.

BASIC-E does not have a SPC (x) function. Use the editor to replace "SPC (" with "TAB (pos +". The new function will then be TAB (POS + x), which has the same effect as SPC (x).

The most serious inconsistency between Microsoft BASIC and BASIC-E is in the area of variable names. Microsoft BASIC allows a variable name used as a dimensioned array to be independently used as a simple variable. For example, Microsoft BASIC would allow the use of A (3, 2) and A as two separate variables in the same program. BASIC-E does not allow variable names used on arrays to be used as simple variables. The solution involves either changing the array variable name or the simple variable name.

BASIC-E is very particular when it comes to print statements. Semicolons must separate all sections of BASIC-E print statements. For example, 100 PRINT "NUMBER" NUM" DATE" DAT would run properly in Microsoft BASIC. The line would have to be connected to the following form to run in BASIC-E:

```
100 PRINT "NUMBER"; NUM; "DATE"; DAT
```

BASIC-E will not insert missing spaces in a line as Microsoft BASIC will. For example:

```
100FORI=1TO10:NEXTI
```

The above line would run in Microsoft BASIC but it would have to be converted to the following form to run in BASIC-E:

```
100 FOR I = 1 TO 10: NEXT I
```

Much has been written on the subject of Conway's game of Life, but most of the literature on this subject has been devoted to configurations, patterns and theorems. Little has been written about methods of actually implementing Life programs on a computer. To correct the paucity of information, this article is devoted almost exclusively to Life algorithms. My associates Peter Raynham and David Buckingham have given invaluable assistance in the development and refinement of the algorithms contained herein. For convenience, the algorithms, with a few obvious exceptions, will be given in a pseudolanguage resembling ALGOL.

Life Algorithms

Mark D Niemiec
Computer Communications Network Group
E4, room 2369A
University of Waterloo
Waterloo, Ontario
CANADA N2L 3G1

Rules of Life

The game of Life deals with patterns that change according to certain rules. The changes proceed in discrete steps called *generations.* The patterns are configurations of bits (or cells) which may have two states: live and dead. These states are analogous to true and false or 1 and 0. The Life cells are arranged in a square field of infinite size (ie: arbitrarily large). Each cell has eight neighbors: those cells orthogonally adjacent to it on all sides as well as diagonally. If a live cell has two or three live neighbors, it lives in the next generation: otherwise it dies. If an empty cell is surrounded by exactly three living cells, a new one is born there in the next generation. Dead cells not meeting this requirement remain empty.

Simple Algorithm

A simple algorithm for performing these operations is shown in listing 1. The pattern resides in the *old* and *new* arrays within limits MINX through MAXX and MINY through MAXY. The major problem with this algorithm is that it requires two arrays, one to hold the present generation, and one to hold the next. Since only one bit is really used in each element of the array, both arrays may be stored together in the computer memory using one byte per element. The cells of the new generation are kept in one half (or nybble) of the byte, while the data for cells of the current generation is stored in the upper half of the byte. This has the advantage of using only one loop, since the loop at the end is no longer required to copy data between arrays.

Improving the Simple Algorithm

A simple observation leading to further improvement is that the computed value of a cell in the next generation will be 1 if and only if the sum of the values of itself and the values of cells in its neighborhood is 3. This pertains to the case where a cell is 1 and the neighborhood is 2, and to the case where a cell is 0 and the neighborhood is 3. The new cell will be 1 only if the neighborhood ORed with the old cell itself is equal to 3. If we use a language with primitives like AND, OR, XOR, and NOT for bit manipulation, we can eliminate one of the IF statements. This makes the code simpler and faster. This new algorithm appears in listing 2.

Each of the above algorithms acts on one cell at a time. While this approach is easy to program, it is not terribly efficient. Even using assembler language we get an efficiency ratio of only about 15 to 20 instructions per bit. This may seem adequate for most applications, but it is not sufficient when running extremely large patterns, or running any pattern for a very long time. While one or two bits per byte may be adequate storage density for many applications, small computers and large patterns demand greater density.

Other Possibilities for Improvement

Let us now examine algorithms which are not restricted to calculating or automating the next generation for one bit at a time. The obvious advantages of such algorithms are increased memory and time efficiency. However, there are several major disadvantages from the point of view of the algorithm designer. Since the program must perform several calculations simultaneously, we cannot use decision primitives (ie: the IF statements in the previous examples).

We are therefore restricted to using

About the Author

Mark Niemiec is currently a third year mathematics student at the University of Waterloo. Introduced to Life by one of the articles in Scientific American, *he has been actively involved in local Life groups in Sarnia and Waterloo, Canada. He has had several Life related discoveries published in the now defunct LIFELINE newsletter. His other related interests include computer graphics and automata theory.*

individual bit and byte manipulations, in much the same way that a parallel adder device adds numbers without making decisions about such things as carry bits. The algorithm in listing 3 uses binary integers of any length, allocating three bits per Life cell. Using this algorithm, we can effectively automate n/3 cells at the same time, where n is the word size in bits.

The entire pattern is divided into vertical strips of n/3 columns, which are all automated simultaneously. The program must include means to provide the necessary neighbors for the cells at the edges of the strips. With the other algorithms, we can keep track of the limits of the pattern and automate only within those limits. However, using the vertical strip method, we can keep track of the height of the pattern only, since the pattern is chopped up across its entire width.

Vertical Strip Method

In the example shown in listing 3, I used a hybrid notation combining high level and lower level instructions in order to concentrate on the actual calculation algorithm. I wished to avoid using low level code for the mechanics of indexing and loops, which are better illustrated in a high level language. I tried to choose low level instructions that are available on most machines: load and store, left and right shifts and Boolean operations.

In the coding of listing 3, the lines beginning at point 1 copy over the bits at the edge of each strip in the overall pattern to the extreme edges of the adjacent strips so that one contiguous array is simulated. Lines starting at point 2 add up the values of the neighbors and perform a logical OR of the current state, as described earlier. Lines of code at point 3 basically calculate the following for every bit:

$$B_R = \text{sum}_{(1\ \text{bit})} \cap \text{sum}_{(2\ \text{bit})} \cap \overline{(\text{sum}_{(4\ \text{bit})})}$$

Or in English, the resultant bit (B_R) is the logical sum of the 1 bit ANDed with the sum of the 2 bit ANDed with the negation of the sum of the 4 bit. The code at point 4 cleans up the garbage bits and stores the new result. Lines of code after point 5 merely copy the new field to the old, as before.

With this algorithm, the interior of the calculation loop takes 17 instructions. Subscripting is assumed to be "free" for array indices so close together, with constant displacements. If we use a processor with a word length of 16 bits, we have an efficiency ratio of 17/4, or less than 5 machine

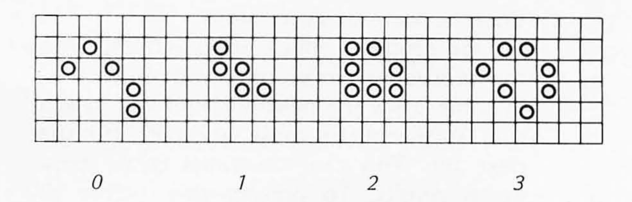


Figure 1: Transformation of a Life pattern through three generations. This process is sometimes referred to as the automation of the pattern. Generation 0 is the original pattern of live cells. The succeeding generations proceed according to the rules of birth, existence and death.

```
FOR  Y := MINY TO MAXY BY 1  DO
BEGIN
     FOR  X := MINX TO MAXX BY 1  DO
     BEGIN
          SUM := OLD(Y-1,X-1) + OLD(Y-1,X) + OLD(Y-1,X+1)
               + OLD(Y  ,X-1)              + OLD(Y  ,X+1)
               + OLD(Y+1,X-1) + OLD(Y+1,X) + OLD(Y+1,X+1);
          IF  SUM = 3  THEN  NEW(Y,X) := 1
          ELSE  IF  SUM = 2  THEN  NEW(Y,X) := OLD(Y,X)
          ELSE  NEW(Y,X) := 0
     END
END

FOR  Y := MINY TO MAXY BY 1  DO
BEGIN
     FOR  X := MINX TO MAXX BY 1  DO
     BEGIN
          OLD(Y,X) := NEW(Y,X)
     END
END
```

Listing 1: The basic algorithm for computing Life generations. The code presented here is written in a pseudolanguage resembling ALGOL. Each element of the matrices represents one cell. The second group of statements copies the complete new generation matrix into the old generation matrix so that the procedure may repeat.

```
FOR  Y := MINY TO MAXY BY 1  DO
BEGIN
     FOR  X := MINX TO MAXX BY 1  DO
     BEGIN
          SUM := OLD(Y-1,X-1) + OLD(Y-1,X) + OLD(Y-1,X+1)
               + OLD(Y  ,X-1);
          SUM := (SUM/16)                  + OLD(Y  ,X+1)
               + OLD(Y+1,X-1) + OLD(Y+1,X) + OLD(Y+1,X+1);
          SUM := SUM OR OLD(Y,X);
          OLD(Y,X) := OLD(Y,X) * 16;
          IF  SUM MOD 8 = 3  THEN  OLD(Y,X) := OLD(Y,X) + 1
     END
END
```

Listing 2: A routine using bit manipulation by the logical OR operation. This enables the program to store the old and new generations in a single matrix. The key point is that a cell in the new generation will be 1 only if the sum of its neighbors ORed with 1 gives the result 3. To code this routine in low level machine instructions, the user may use left and right shift operations to perform the multiplication and division.

instructions per bit, which is much better than the previous algorithms.

One problem which arises is that, since the procedure works only with three bits, in some cases the neighborhood will have a sum sufficient to cause a carry into the next bit. This can sometimes create erroneous results. To prevent this, before all eight neighbors have been added together, the sum must be thus folded: if $sum_{(4\ bit)}$ is on, then turn off $sum_{(1\ bit)}$ and $sum_{(2\ bit)}$.

```
          FOR   Y  := MINY  TO MAXY  BY 1    DO
          BEGIN
[1]            OLD(Y,MINX-1)  := OLD(Y,MAXX)  /  8
               OLD(Y,MAXX+1)  := OLD(Y,MINX)  *  8
               FOR   X  := MINX  TO MAXX  BY 1   DO
               BEGIN
[2]                 LOAD      OLD(Y-1,X-1)
                    ADD       OLD(Y-1,X  )
                    ADD       OLD(Y-1,X+1)
                    ADD       OLD(Y  ,X-1)
                    ADD       OLD(Y  ,X+1)
                    ADD       OLD(Y+1,X-1)
                    ADD       OLD(Y+1,X  )
                    STORE     TEMP1
                    NOT
                    AND       OCTAL'444...4'
                    RIGHT     1
                    STORE     TEMP2
                    RIGHT     1
                    OR        TEMP2
                    AND       TEMP1

                    ADD       OLD(Y+1,X+1)
                    OR        OLD(Y  ,X  )
[3]                 STORE     TEMP1
                    NOT
                    RIGHT     1
                    AND       TEMP1
                    RIGHT     1
                    AND       TEMP1
[4]                 AND       OCTAL'1111....1'
[5]                 STORE     NEW(Y  ,X  )
               END
          END

          FOR   Y  := MINY  TO MAXY  BY 1    DO
          BEGIN
               FOR   X  := MINX  TO MAXX  BY 1   DO
               BEGIN
                    OLD(Y,X)  := NEW(Y,X)
               END
          END
```

Listing 3: Routine to compute Life generations using cell data packed with multiple cells in a word. Several cells can have their new generations computed, or be automated at the same time. The code here is a generalized machine language with the high level iteration instructions retained for clarity. The packing density is n/3, where n is the word size in bits. The ellipses in the octal constants indicate that the method works for any word size. This method may be referred to as the vertical strip method from the manner in which it divides up the Life universe.

Complete Storage Efficiency

All of the cell storage methods previously described involve wasting from 75 to 87 percent of storage in order to maintain program simplicity. We must eventually ask, "Why not use *every* bit?" By completely packing the data, cell storage becomes about eight times as compact as the original algorithm allowed.

When each bit is surrounded by relevant data, though, a drawback arises: there is no room for any kind of arithmetic instructions which produce a carry to the next bit. We remove this difficulty by performing the arithmetic in the same way that hardware performs it. Since there exists no semiconductor "add" gate, one adds by selectively combining the AND and exclusive OR (XOR) gates. Similarly, here it is possible to add using only a large assortment of logical operators.

One such solution, using PDP-11 machine code, is shown in listing 4. The BIC instruction of the PDP-11, which is effectively an AND/NOT instruction, is actually more useful then a normal AND instruction in this case. A similar implementation for a different processor not having this instruction might be slightly larger.

This example shows that we can automate 16 bits at a time in about 40 instructions of 16 bits, which is about 2 1/2 instructions per bit – more efficient than any previous method. If 8 bit instructions are used, we must approximately double the number of instructions per cell. This also holds for the previous algorithms of many bits per word. As an example of how this improves performance, we are able to automate 72 bits at a time on the Honeywell 66/60 by looping with double word instructions with an average of almost 2 bits per instruction. So, despite the fact that this method has the longest code segments inside the calculation loop, it is by far the most efficient method of those discussed.

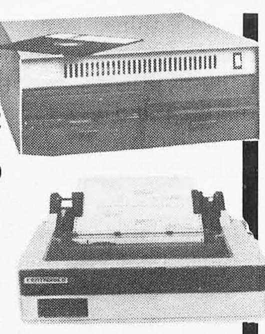
Life in Three States

One generalization of Life is a 3 state version called *Immigration,* which has the regular blank or dead state, plus *two* living states, X and O, for instance. The rules are the same as those for Life, except that, upon birth, a new cell's state is determined by the majority of parent cells. For example, three X cells or two X and one O will produce an X offspring. One use of the two distinct living states is to distinguish the major predecessor of an object. A case in point: if two gliders collide to form a block, from which one does the block actually originate?

HOW TO BEAT THE SYSTEM WITH SYNCHRO-SOUND

High-performance, low cost complete Microcomputer Business Systems (plus Software) at EXTRA savings only Synchro-Sound can offer!

SYSTEM No. 1



DIGITAL SYSTEMS DSC-2 Z-80 COMPUTER SYSTEM

32K RAM, 4 RS-232 serial interfaces, 16 bits of parallel I/O. Double density floppy disk, Shugart dual-drive,
Regular price $4995.00

CENTRONICS 700 PRINTER With RS-232 Interface

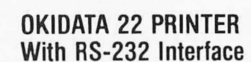
60 characters per second, 132 column line, tractor feed.
Regular price $1720.00

HAZELTINE 1500 VIDEO TERMINAL

80 character lines, 24 line page, line and page editing. Separate numeric keypad.
Regular price $1149.00
$7864.00

THE COMPLETE SYSTEM ONLY **$7495.00**
SAVE $369.00!

SYSTEM No. 2



IMSAI VDP 80 DATA PROCESSING COMPUTER SYSTEM

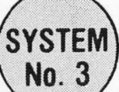
32K RAM, expandable to 196K. Parallel and serial I/O. Megabyte mass storage, alphanumeric intelligent keyboard.
Regular Price $6995.00

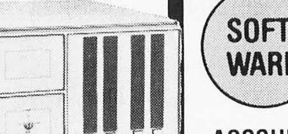
OKIDATA 22 PRINTER With RS-232 Interface

Tractor feed, 132 column, 125 lpm, upper and lower case. 12 different fonts on command.
Regular price $2888.00
$9883.00

THE COMPLETE SYSTEM ONLY **$9445.00**
SAVE $438.00!

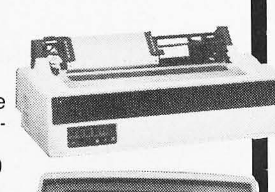
SYSTEM No. 3





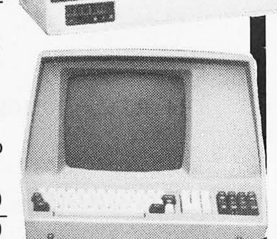
CROMEMCO SYSTEM 3 Z-80 COMPUTER SYSTEM

32K RAM, dual disk drive, RS-232 Interface. S-100 bus.
Regular price $5990.00



MODEL 3703 LINE PRINTER

180 characters per second, print line width 132 cols., bi-directional printing.
Regular price $2995.00



CRT TERMINAL MODEL 3100

80 characters per line, 24 lines, up to 19,200 baud, upper and lower case.
Regular price $1595.00
$10,580.00

THE COMPLETE SYSTEM ONLY **$10,095.00**
SAVE $485.00!

SOFTWARE

Three software packages* that enable you to profitably make full use of your Microcomputer Business System

ACCOUNTS RECEIVABLE PACKAGE Creates transaction file of charges, credits and payments. Batch mode for monthly statements and aged schedules.
Operators manual and disk, single copy **$500.00**

INVENTORY PACKAGE Maintains current listing of stock items, master inventory listing with price and cost data.
Operators manual and disk, single copy **$500.00**

ORDER-ENTRY PACKAGE (Requires Accounts Receivable and/or Inventory Package) Links Accounts Receivable and Inventory packages into one coherent, easy-to-use, time-saving system.
Operators manual and disk, single copy **$500.00**

Buy any of our 3 MICROCOMPUTER SYSTEMS and any combination of these 3 SOFTWARE PACKAGES ARE YOURS FOR

HALF PRICE!

*Require 48K Memory and Microsoft Basic

Terminal and Printer substitutions may be made on any of the above Systems

CUSTOMIZED HARDWARE SYSTEMS ARE AVAILABLE

Complete modern servicing and consulting facilities on premises



SYNCHRO-SOUND ENTERPRISES, INC.
The Computer People
193-25 Jamaica Avenue,
Jamaica, New York 11423
212/468-7067 TWX: 710-582-5886
Hours 9-4 daily and Saturday
Visit our new showroom
Working units on display
Dept BSY
BankAmericard • Master Charge

Circle 355 on inquiry card.

The major difference between this algorithm and regular Life is that one must keep track of the kind of bit, as well as whether a bit is alive or not. An algorithm which does this is shown in listing 5. For a living cell of type X, bit 1 of the byte is set to 1; for a type O live cell, both bit 1 and bit 8 of the byte are set to 1. The distinctive feature of this algorithm is that when a new cell is created, the kind of cell it becomes is determined by how many 1s from bit 8 have been added into the sum. Since bit 8 is uniquely used, packing the old generation and the new into two halves of the same byte is no longer possible. Note that this algorithm will work in the same manner as a standard Life algorithm on an array containing only one of the living states.

Life Generations of the Fourth Kind

An extension of Immigration results in a version of Life having four different kinds of living states. The kind of cell which

```
MOV     OLD(Y-1,X-1),R0     (R1,R0) = NEIGHBOURS 1+2
MOV     OLD(Y-1,X  ),R1
XOR     R1,R0
BIC     R0,R1

MOV     OLD(Y-1,X+1),R2     (R2,R0) = NEIGHBOURS (1+2)+3
XOR     R2,R0
BIC     R0,R2
BIS     R2,R1               (R1,R0) = NEIGHBOURS 1+2+3

MOV     OLD(Y+1,X-1),R2     (R2,R3) = NEIGHBOURS 4+5
MOV     OLD(Y+1,X  ),R3
XOR     R3,R2
BIC     R2,R3

MOV     OLD(Y+1,X+1),R4     (R4,R2) = NEIGHBOURS (4+5)+6
XOR     R4,R2
BIC     R2,R4
BIS     R2,R3               (R3,R2) = NEIGHBOURS 4+5+6

XOR     R2,R0               (R2,R0) = NEIGHBOURS (1+2+3)+(4+5+6)
BIC     R0,R2

XOR     R2,R1               (R2,R1) = CARRIES (1+2+3)+(4+5+6)
BIC     R1,R2
XOR     R3,R1
BIC     R1,R3
BIS     R3,R2

MOV     OLD(Y  ,X-1),R3     (R4,R3) = NEIGHBOURS 7+8
MOV     OLD(Y  ,X+1),R4
XOR     R4,R3
BIC     R3,R4

XOR     R3,R0               (R3,R0) = NEIGHBOURS (1+2+3+4+5+6)+(7+8)
BIC     R0,R3

XOR     R1,R3               (R4,R1) = CARRIES (1+2+3+4+5+6)+(7+8)
BIC     R3,R1
XOR     R1,R4
BIC     R4,R1
BIS     R3,R2
BIS     R4,R2

BIS     OLD(Y  ,X  ),R0     (R0) IS OR'ED WITH ORIGINAL BIT

BIC     R2,R0               NEIGHBOURHOOD OF ≥ 4 CAUSES DEATH
COM     R1
BIC     R1,R0               NEIGHBOURHOOD OF ≥ 2 A NECESSITY
MOV     R0,NEW(Y  ,X  )
```

Listing 4: A program to manipulate cells stored as a single bit. This code uses the instruction set of the Digital Equipment Corp PDP-11 processor. The addition is performed by selective use of logical operations. Application of this method results in the greatest possible storage density. Although there are more lines in the program, automation of many cells simultaneously makes this the most efficient procedure discussed.

Listing 5: A program to perform calculations for the 3 state, or Immigration game. Live cells are of two distinct types. Cells are stored one per byte. For one type of live cell, the byte has its 1 bit set to 1. The other type of live cell is indicated by setting both bit 1 and bit 8 of the byte to 1. Data stored by this scheme may not be packed further.

```
FOR  Y := MINY TO MAXY BY 1  DO
BEGIN
    FOR  X := MINX TO MAXX BY 1  DO
    BEGIN
        SUM := OLD(Y-1,X-1) + OLD(Y-1,X) + OLD(Y-1,X+1)
             + OLD(Y  ,X-1)               + OLD(Y  ,X+1)
             + OLD(Y+1,X-1) + OLD(Y+1,X) + OLD(Y+1,X+1);
        IF  OLD(Y,X) ¬= 0  THEN
            IF  (SUM OR 1) MOD 8 = 3  THEN  NEW(Y,X) := OLD(Y,X)
            ELSE  NEW(Y,X) := 0
        ELSE  IF  SUM MOD 8 = 3  THEN
            IF  SUM < 16  THEN  NEW(Y,X) := 1
            ELSE  NEW(Y,X) := 9
        ELSE  NEW(Y,X) := 0
    END
END

FOR  Y := MINY TO MAXY BY 1  DO
BEGIN
    FOR  X := MINX TO MAXX BY 1  DO
    BEGIN
        OLD(Y,X) := NEW(Y,X)
    END
END
```

Listing 6: A program that demonstrates a version of Life in which cells are in any one of five possible states. A hash table is used to determine the kind of cell which is born to various sets of parents.

```
ARRAY HASH(0::33) CONTAINING
    1,  1,  9,  9,  0,  0,  0,  1,  89, 9,
    0,  1,  57, 9,  57, 57, 0,  0,  9,  1,
    0,  57, 89, 89, 0,  57, 0,  0,  0,  89,
    0,  0,  0,  89;

    FOR  Y := MINY TO MAXY BY 1  DO
    BEGIN
        FOR  X := MINX TO MAXX BY 1  DO
        BEGIN
            SUM := OLD(Y-1,X-1) + OLD(Y-1,X) + OLD(Y-1,X+1)
                 + OLD(Y  ,X-1)               + OLD(Y  ,X+1)
                 + OLD(Y+1,X-1) + OLD(Y+1,X) + OLD(Y+1,X+1);
            IF  OLD(Y,X) ¬= 0  THEN
                IF  (SUM OR 1) MOD 8 = 3  THEN  NEW(Y,X) := OLD(Y,X)
                ELSE  NEW(Y,X) := 0
            ELSE  IF  SUM MOD 8 = 3  THEN  NEW(Y,X) := HASH(SUM/8)
            ELSE  NEW(Y,X) := 0
        END
    END

    FOR  Y := MINY TO MAXY BY 1  DO
    BEGIN
        FOR  X := MINX TO MAXX BY 1  DO
        BEGIN
            OLD(Y,X) := NEW(Y,X)
        END
    END
```

Listing 7: The game of Life programmed in one line of APL. This function accepts the input pattern as a Boolean matrix and displays it every generation. The reader is left to determine which algorithm is used.

```
      ∇ LIFE X ;Y
[1]     →1,ρ⎕←' ⎕'[X←(3=Y)∨X∧4=Y←(¯1⌽Y)+Y+1⌽Y←(¯1⊖X)+X+1⊖X←
        (0,((1↑ρX)ρ1),0)⍀X←(0,((1↓ρX)ρ1),0)\X←
        (((⍳1↓ρX)≥(∨⌿X)⍳1)∧⌽(⍳1↓ρX)≥(⌽∨⌿X)⍳1)/X←
        (((⍳1↑ρX)≥(∨/X)⍳1)∧⌽(⍳1↑ρX)≥(⌽∨/X)⍳1)⌿X]
      ∇
```

may be born is determined by the majority of the parent cells. If no majority exists (ie: all three parents are different), a cell of the fourth kind is generated. Patterns containing only two kinds or one kind of live cells make this game become equivalent to Immigration and Life, respectively.

Listing 6 shows a procedure for automating this Quad-Life. It is essentially the same as the Immigration algorithm, except that the determination of cell type is done most efficiently by using a hash table. The program may index a table containing cell types using the sum of the types of the parent cells. Since a birth can occur only between three live cells, the hashing function need only meet the condition that it be unique when any combination of three of the four different values representing the four different cell types is added together. The values 1, 9, 57, and 89, produce a unique hash. This is also the smallest solution, giving a hash table with 34 entries, of which 20 are actually needed.

Exploiting Language Power

No proper treatment of algorithms is complete without a program written in one line of APL which performs the desired task. The program in listing 7 accepts the input pattern as a Boolean matrix and displays the pattern every generation. The grid is effectively of infinite size, since storage allocation for the pattern is grown and shrunk as necessary. The program works only in origin 0, but changing it to work in origin 1 would be fairly trivial. Determining which algorithm is used here will be left as an exercise to the reader.

Ultimate Purpose

The more research that is done concerning Life, the more it appears that investigators are becoming less concerned with merely empirical questions such as "What does pattern such and such eventually form?" — and more concerned with more esoteric problems such as entropy, information theory and computability theory. Due to this shift in emphasis, it is becoming apparent that the future of Life algorithms will not rest in smaller and faster programs to automate Life. The advances of the future abide rather in highly specialized abstract mathematical programs for proving theorems or gathering statistics. However, for most enthusiasts, Life will always be a fascinating diversion that produces fascinating pictures on the screens of video displays.■

Making a list?

DR DOBB'S JOURNAL OF COMPUTER CALISTHENICS AND ORTHODONITA, volume I edited by Jim C Warren.

☐ This is a collection of everything from the first year of Jim Warren's outstanding journal. 359 pp. $13.00 (no photo).

YOUR HOME COMPUTER by James White.

☐ A clearly written nontechnical description of personal computers that required no prior knowledge of computers or electronics. Yours for $6.00.

GETTING INVOLVED WITH YOUR OWN COMPUTER by Leslie Solomon and Stanley Viet.

☐ Whether your interest in business applications, word processing, education, security, computerized video games, computer art and music. Robots, or amateur radio, this book will get you started and keep you on the right track. 216 pp. $5.95.

UP YOUR OWN ORGANIZATION! by Donald M Dible.

☐ A great handbook on how to start and finance a new business, this is the most comprehensive reference we've seen on the subject. 372 pp. Available for $14.95 in hardcover.

HOW TO PROFIT FROM YOUR PERSONAL COMPUTER by Ted G Lewis.

☐ If you have wanted to use your personal computer in a profitable manner, and you feel you need some help planning programming techniques for common business applications, this book is for you. 191 pp. $7.95.

BASIC AND THE PERSONAL COMPUTER by Thomas A Dwyer and Margot Critchfield.

☐ After giving an in-depth course in BASIC, the book discusses microcomputer hardware, graphics, word processing, sorting, simulation and data structures. Easy to read and informative. 438 pp. $12.95 (no photo).

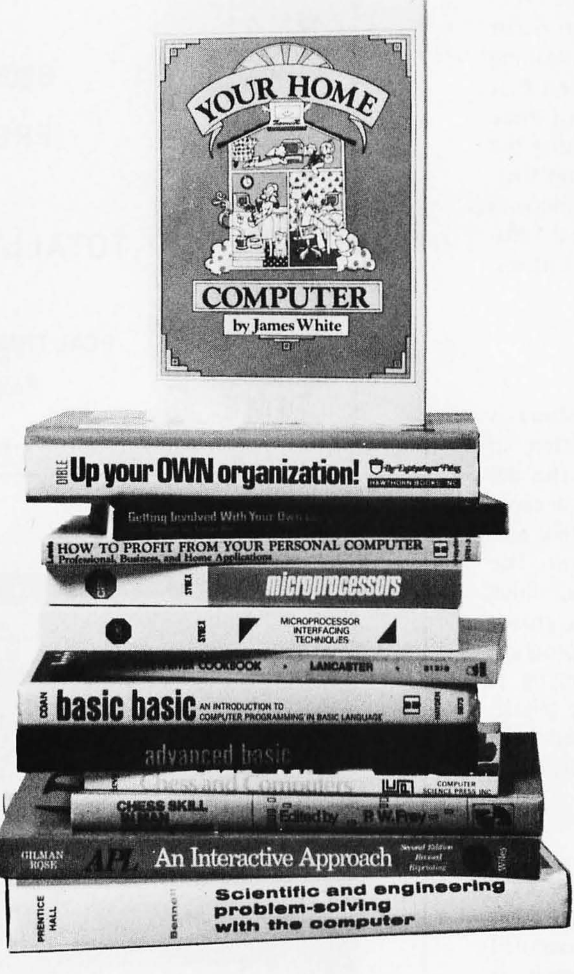
TV TYPEWRITER COOKBOOK by Don Lancaster.

☐ A complete guide to low cost television display of alphanumeric data, several chapters of which were published ahead of the book in early issues of BYTE magazine. $9.95.

THE CHEAP VIDEO COOKBOOK by Don Lancaster.

☐ This latest in Don Lancaster's series of hardware books continues where the **TV Typewriter Cookbook** leaves off. New, inexpensive video display circuits (and the software to drive them) are described. 256 pp. $5.95 (no photo).

Check here for new ideas



APL AN INTERACTIVE APPROACH Second Edition, Revised by Gilman and Rose.

☐ Here's an excellent way to introduce yourself to the APL language. Gilman and Rose have extensively updated their popular book to include the latest information about APL and the various forms now in use. $11.95.

SARGON by Dan and Kathe Spracklen.

☐ Written about the 1978 West Coast Computer Faire chess champ, "Sargon," highly praised, is a program in Z-80 assembler language. $14.95 (no photo).

THE WIRED SOCIETY by James Martin.

☐ This is a fascinating and not-to distant account of what the future might be like, changed by the new telecommunications technology. Prefacing his discussions with historical as well as contemporary references, Martin lures the reader into an impending technical revolution, which will change corporations, change government, and changes the lives of everybody. $12.95 hardcover (no photo).

MICROPROCESSOR INTERFACING TECHNIQUES (Expanded 2nd edition) by Austin Lesea and Rodnay Zaks.

☐ Now available in this newly revised edition featuring: new material on "useful circuits,"an enlarged floppy disk section with new double density controllers, updates on LSI controllers—and much more. Including basic concepts and techniques, this book is a very comprehensive handbook on interfacing. 416 pp. $9.95.

MICROPROCESSORS: FROM CHIPS TO SYSTEMS by Rodnay Zaks.

☐ A complete and detailed introduction to microprocessors and microcomputer systems. $9.95.

BASIC BASIC, 2nd EDITION by James S Coan.

☐ If you're not already familiar with BASIC, this book is one of the best ways to learn about this popular computer language. BASIC (which stands for Beginner's All-Purpose Symbolic Instruction Code) is easy to learn and easy to apply to many problems. $8.95.

ADVANCED BASIC by James S Coan.

☐ This is the companion volume to James Coan's **Basic BASIC.** In this book you'll learn about some of the more advanced techniques for programming in BASIC, including string manipulation, the use of files, plotting on a terminal, simulation and games, advanced mathematical applications and more. $8.95.

CHESS AND COMPUTERS by David Levy.

☐ This 145-page paperbound book is loaded with chess games played by computers. The games are computer versus computer and computer versus human. $8.95.

CHESS SKILL IN MAN AN MACHINE edited by Peter W Frey.

☐ This is a most fascinating book, concerning itself with the when, how, and why of computer chess. $14.80 hardcover.

SCIENTIFIC AND ENGINEERING PROBLEM SOLVING WITH THE COMPUTER by William Ralph Bennett Jr.

☐ Besides teaching BASIC (which it does admirably), this lively, lucid book presents a wealth of imaginative and unusual applications programs taken from many disciplines. In its diversity and elegant style, it ranks with Donald Knuth's works as a milestone in the art of computing. Hardcover, $19.95.

GAMES, TRICKS AND PUZZLES FOR A HAND CALCULATOR by Wallace Judd.

☐ Got a friend who won't have anything to do with calculators? You just might woo him or her into the electronic age with some of the tricks in this book. Capers, games and astounding calculations are here to dazzle. They've even pulled a calculator apart to show you what the insides look like! A neat book for anyone suffering from technophobia or plain boredom. 91 pp. $4.95.

FUNDAMENTALS OF RECORDKEEPING AND FINANCE FOR THE SMALL BUSINESS by Robert C Ragan, CPA, and Jack Zwick, Ph.D.

☐ Once you have your organization or business up and running, records must be kept. 196 pp. Only $10.00 in hardcover.

THE ART OF COMPUTER PROGRAMMING by Donald E Knuth.

Praised by many critics as the best books in their field. These books are part of a projected seven volume omnibus survey of computer science now being completed by Donald E Knuth.

☐ **Volume I, Fundamental Algorithms,** 634 pp. $21.95 hardcover.

☐ **Volume II, Seminumerical Algorithms,** 624 pp. $21.95 hardcover.

☐ **Volume III, Searching and Sorting,** 722 pp; $21.95 hardcover.

FUNDAMENTALS AND APPLICATIONS OF DIGITAL LOGIC CIRCUITS, 2nd edition by Sol Libes.

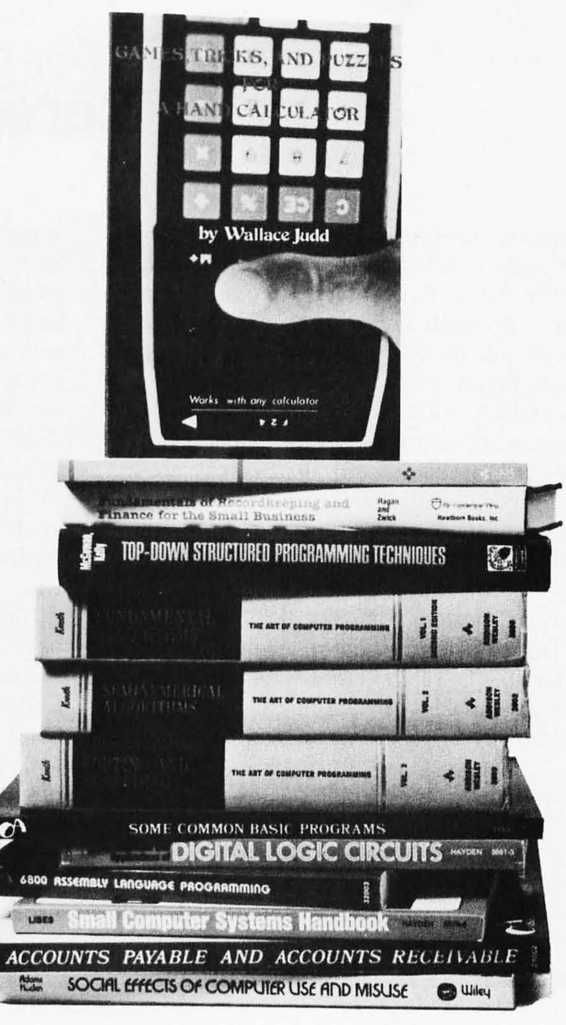
☐ You'll find everything from the basic theory, to various circuit designs, to the most recent applications, including the circuitry of calculators, digital voltmeters, frequency counters, and computers. 200 pp. $7.95.

SMALL COMPUTER SYSTEMS HANDBOOK by Sol Libes.

☐ This is an introductory text for those with a minimal knowledge of electronics who are interested in microcomputers as a hobby. The author explains microcomputer concepts from number systems and logic circuits to programming languages and applications with the emphasis on practical knowledge. 197 pp. $8.45.

THE LITTLE BOOK OF BASIC STYLE by John M Nevison.

☐ This book teaches you how to write clear, well-planned programs you can read. 151 pp. $5.95 (no photo).



SOME COMMON BASIC PROGRAMS by Lon Poole and Mary Borchers.

☐ At last, a single source for all those hard to find mathematics programs! All the programs are written in a restricted BASIC suitable for most microcomputer BASIC packages, and have been tested and debugged by the authors. $8.50.

SOCIAL EFFECTS OF COMPUTER USE AND MISUSE by J Mack Adams & Douglas H Haden.

☐ Most of us have come to accept the presents of computers in our lives. Their effect on society as a whole is worthy of carefull consideration. This book surveys the evolution of computers, their application, and the social, economic and philosophical implications of their increasing use. 326 pp. $10.00.

PROGRAMMING A MICROCOMPUTER: 6502 by Caxton C Foster.

☐ This informal, well-written book may be just what you need to enter the world of microcomputers. Foster uses the 6502 microprocessor (used in the KIM-1, the PET, and the Apple microcomputer systems) as a basis for discussing the techniques of writing programs. 231 pp. $9.95.

TOP-DOWN STRUCTURED PROGRAMMING TECHNIQUES by Clement L McGowan & John R Kelly.

☐ What is structured programming? The authors answer this question in this lively, well-written book. 288 pp. $15.95 hardcover.

6800 ASSEMBLY LANGUAGE PROGRAMMING by Lance A Leventhal.

☐ From Osborne & Associates we have another in their series of microcomputer books. Lance A Leventhal discusses assembly language programming for computers using the 6800 microprocessor. The 6800 instruction set is presented in depth with chapters on assemblers, simple programs, code conversion, tables and lists, subroutines, input/output, interrupts, program design and documentation, and sample projects. $8.50.

ACCOUNTS PAYABLE AND ACCOUNTS RECEIVABLE by Lon Poole and Mary Borchers.

☐ This is a set of 21 programs which constitute an invoice-linked accounts payable and a low-volume invoice accounts receivable system for small businesses. It is the third in Osborne & Associates' series of BASIC business programs and will be complemented by the soon to be published General Ledger. 318 pp. $15.00.

BITS ™ inc

Books to erase the impossible

POB 428, 25 Route 101 West
Peterborough, NH 03458

DIAL YOUR BANK CARD ORDERS ON OUR **TOLL-FREE HOT LINE: 1-800-258-5477.**

Name ____________________

Address ____________________

City ____________ State ________ Zip ________

Signature ____________________

master charge | BANKAMERICARD | VISA

My check enclosed:

Check #: ____________________

Amount: ____________________

My card #: ____________________

Expires: ____________________

		Total
Number of books	________	________
Postage $.75/book or $1.00 outside U.S.		________
Grand Total		________

Prices subject to change without notice. *Dealer Inquiries Invited.* *You may photocopy this page.*

The Digicast System

Receiving Data and

A I Halsema
4921 Patrae St
Los Angeles CA 90066

Imagine turning on your radio and receiving the latest copy of BYTE—or receiving only the news of interest to you. Tuning in to a program could mean exactly that: you would receive software source code listings, binary object code listings, and documentation. All of these are possible with a revolutionary new system called Digicast.

Digicast is similar to Britain's Teletext and France's Antiope systems, but would be transmitted by commercial FM radio stations. Teletext data is broadcast during a portion of the vertical blanking time by television stations of the BBC (see figure 1). Decoded by logic connected to the television set, Teletext news and announcements are dialed up and displayed on the television screen at the viewer's option.

Operation

To understand how Digicasting would work, consider FM stereo broadcasts. FM stereo is transmitted as two separate signals modulating the same radio frequency carrier. Both channels of the stereo signal are combined into one monaural signal. This combined signal is what you hear when listening with a monaural only FM receiver. A second signal of 38 kHz away from the center frequency, amplitude modulated by left channel minus right channel difference information, is also transmitted (see figure 2). This is called the pilot carrier, and is used in the receiver's stereo demultiplexer to separate the two stereophonic signals. Stereo and other Subsidiary Communications Authorization (SCA) signals are transmitted only by FM stations due to the wide bandwidth assigned to the stations by the Federal Communications Commission. This wide bandwidth (150 kHz, with deviations of 75 kHz above and below the center frequency of the carrier) is required to reproduce high fidelity music. Stereo transmissions do not exhaust the bandwidth available to an FM station, however. Digicast signals can be broadcast in exactly the same way as stereo-

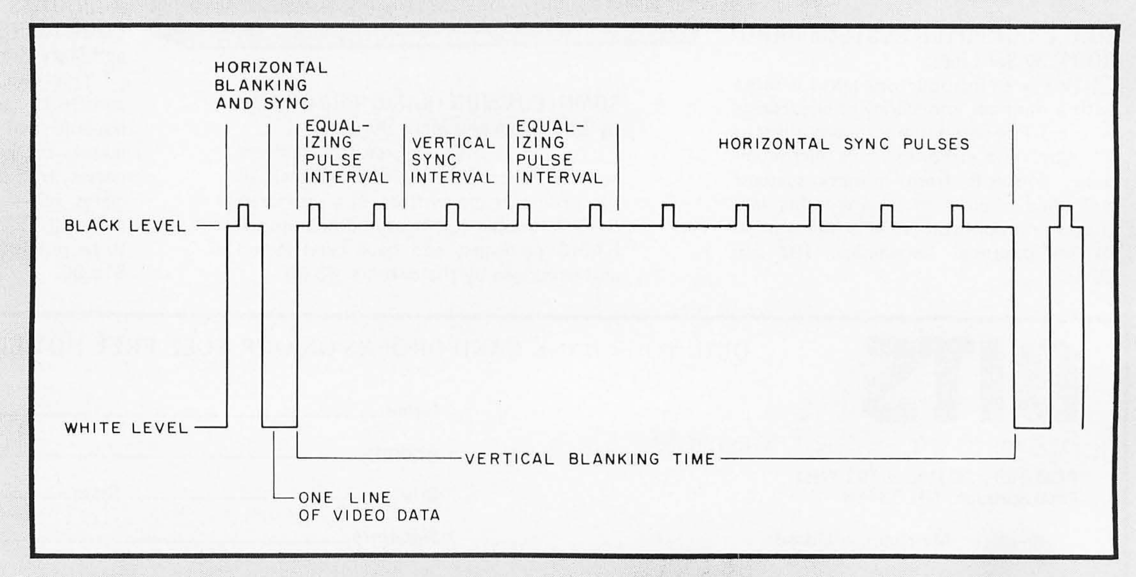


Figure 1: Video synchronization signals. The pulses riding on the waveform at the "blacker than black" voltage level are horizontal sync pulses. Intervals between the horizontal sync pulses can be used to carry information, and are so used in the Teletext and Antiope systems. This portion of the video can be seen if the vertical hold control on the receiver is misadjusted slightly to roll down the picture. Nonblack portions visible in the bar displayed are test signals added at the transmitter. Note that this figure is not to scale, since more horizontal sync pulses actually appear during the vertical blanking time than are shown here.

Information Over Your FM Radio

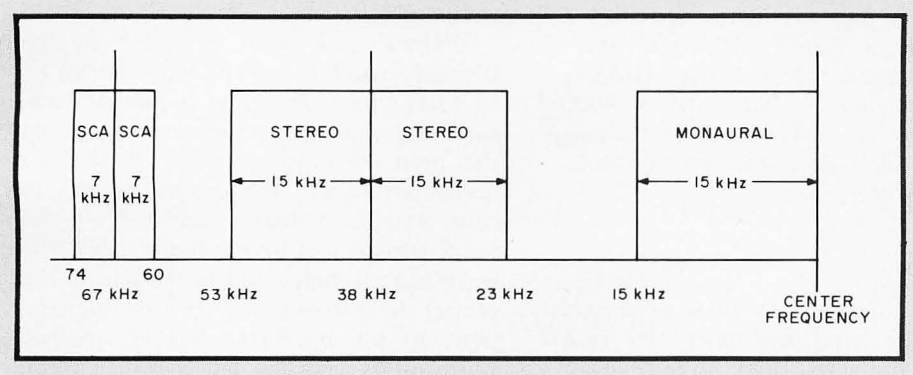


Figure 2: Frequency assignments in an FM station's signal. Not all stations transmit both stereo and SCA (background music). Spaces between the various signals are required to allow separation and reduce mutual interference at the receiver.

Digicast and Digicasting are proprietary names protected by trademark laws.

FM TRANSMITTER
STEREO
OTHER SCA
DIGICAST DATA
WIRE SERVICE TELETYPE; STOCK QUOTATION SERVICE; LOCAL MESSAGE SYSTEM
FM RECEIVER
DIGITAL SIGNAL DEMULTIPLEXER AND AMPLIFIER
UART
COMPUTER
MASS STORAGE
PRINTER
TERMINAL

Figure 3: Simplified block diagram of the Digicast system from input to output.

phonic music. Reception would require the use of a wide bandwidth FM receiver with subcarrier output jacks to allow connection of the Digicast demultiplexer. This demultiplexer would be similar to a standard stereo demultiplexer, but with a different operating frequency and output voltage levels. The demultiplexed data would be fed into a computer via a standard Universal Asynchronous Receiver Transmitter (UART) circuit (see figure 3). Data rates of 4800 to 9600 bps are envisioned. The higher data rates may require the use of special fast rise time receivers.

Applications

Digicast is designed to allow serial read only access to large data bases. One such data base is the news. Used in the newscast mode, Digicast would provide high-speed transmission and reception of all the news in alphanumeric form. The computer at the receiving end would be programmed to select those items containing keywords of interest to its owner and to store or print them for later reading. A news transmission facility is simple to implement, since the news is received in machine intelligible form from the wire services (ie: UPI, AP, Reuters, etc). Software at the receiver can range from the simplest search and save program to complex artificial intelligence programs to correlate data and extrapolate stock market trends. (Stock market and commodities quotations are also available in machine readable form.) As the amount of news and information increases to the saturation point, a UHF television channel (6 MHz bandwidth) could be allocated for this purpose. A 6 MHz channel could handle a 6,000,000 bps data rate, spurring development of still faster microprocessors. An alternate approach to the problem of information saturation is to use additional FM stations in the area to handle additional data. Digicasting will reduce the amount of paper needed for newsprint, and reduce pollution levels. How much of your newspaper do you read? Once read, most newspaper becomes trash, adding to pollution.

Current Status

For Digicast to be a success technically, it must also be a financial success. An FM station can be equipped with the needed electronics for under $10,000, and subscriptions to the service sold to the public on a monthly basis. Specialized Digicast receivers could be mass-produced now for $50, exclusive of the computer, related software and peripherals. To insure that no pirating of the broadcasts takes place, the data can be encrypted, with a new decrypting key being mailed to subscribers on a monthly basis.

Digicasting is being developed by the Digicast Project directed by Jim C Warren Jr, who has edited *Dr Dobb's Journal of Computer Calisthenics & Orthodontia*, and has also been the organizer of the West Coast Computer Faires. The project expects to have a Digicast facility operating in the San Francisco area by the end of 1979, but needs both technical and financial help. The project is currently financed by subscriptions to the *Intelligent Machine Journal*, which prints news and details of the project; you may offer technical help and buy subscriptions ($18 a year) by writing to Jim C Warren Jr, 345 Swett Rd, Woodside CA 94062.■

REFERENCES

1. "Bandwidth Requirements for FM," (no author cited), *Electronics World*, volume 80, number 3, September 1968, page 27.

2. Grob, Bernard, *Basic Television*, second edition, McGraw-Hill, New York, 1964, pages 68 and 108 thru 113. See this reference for frequency modulation terms and television signal waveforms.

3. Solomon, L, "Hidden TV Test Signals," *Electronics World*, volume 71, number 4, April 1964, page 34.

4. Warren, Jim C, *Computer*, IEEE Computer Society Journal, September 1978, page 97. A short biography of Mr Warren is provided on this page.

5. Warren, Jim C, "The Digicast Systems," *Dr Dobb's Journal of Computer Calisthenics & Orthodontia*, volume 3, number 9, October 1978.

6. "Will Video-Text Systems Travel Well?" (no author cited), *Electronics*, volume 51, number 19, September 1978, page 24. This is a very pessimistic editorial on Teletext and personal computing.

7. Winfree, Robert W, "SCA Background Music Multiplexer," *Electronics World*, volume 70, number 6, December 1963, page 42.

Write bug-free code faster in BASIC, FORTRAN, or COBOL with our unique software development system

Enjoy a bonus in better documentation and easier modification, too

Yes, you really can get flawless code faster, using the Stirling/Bekdorf™ system of software development tools with structured programming concepts. Not only does the system help prevent errors, it also helps you trap errors you do make, quickly.

Each tool is human-engineered to do its job better. Every step in the programming process, from initial logic design through print-out or display layout, flows more smoothly with the Stirling/Bekdorf system. Designed to make fullest use of structured programming concepts, our software development tools are human-engineered to reduce initial errors, improve de-bugging speed, and aid concept communication.

When you use "top-down" structured programming, the first part of our system to help you is the 78F2 Flowchartrix™, no matter what language you're using. You use it first to lay out your original concept blocks. Then you use it to write a finely detailed flowchart quite smoothly.

54% more logic cells than other flowchart forms, so you get far more of your program on each page. Each Flowchartrix has a full 77 logic cells, not just 50. This not only saves paper, but also makes your finished flowcharts easier to understand. By seeing up to 27 extra steps of your program on each page, you comprehend program flow more clearly. That's important while writing the flowchart, more important when you write actual code. It's also extremely helpful when you debug, and indispensable when you come back months or years later to modify your original work. 78F2's higher matrix count makes your flowcharts quicker to debug because there are fewer pages to search for errors. Fewer pages also save you money and storage space.

Unique matrix can show your loops AS loops. The 7 x 11 matrix gives you a central column to use for executive control logic, plus 3 columns on each side to describe loops and subroutines. In fact, these 3 columns on each side give plenty of room to write loops laid out visually as loops (sort of a squared circle). This makes loops and subroutines easier to recognize, because their form is readily apparent at a glance. Since they're easier to find, and may even be completed on a single page, they're also simpler to debug or modify.

Every matrix cell has a specific label to help you track branch points. Now it's far easier to follow your program from page to page, point to point. When you write program documentation, having a separate reference point for each cell makes your program much easier to describe clearly.

With Flowchartrix, you don't need a shape template to draw remarkably regular logic symbols. Guides for the most-used logic symbols are right in each matrix cell. They help you draw most standard flowchart symbols entirely free-hand. With 78F2, your pencil can fly as fast as you can write, without interruption. When a flash of insight strikes, now you can keep your pen on paper, flowing rapidly from one step to the next without breaking to pick up a template.

78P4 Print-Out Designers are the next step. When you finish flowcharting, you're ready to lay out the printed reports your program will generate. This helps you focus your thinking on the end result to be produced from all your effort. Then when you begin to follow the flowchart to write actual code you'll be ready to blaze through the report generation segments right along with the sorts and computations.

Unique 70 x 160 matrix accommodates even proportional-spacing word processor formats. The 160-column width can handle practically any printer format. The 78P4 is big, 14½ x 22 inches, because we've scaled the cell size to human writing comfort, not machine print. Each 78P4 character cell is 3mm wide by 5mm high, giving nearly twice the character-writing area of other printout design sheets. Special markings at columns 80, 96, and 132 show you the relative position of each printed character. Scaled page length arrows reference the last possible printed line for standard 8½" and 11" paper (for printers with 6 vertical lines per inch).

Special 5-column area records the program line number of the code which creates each printed line. This unique feature makes the control code much faster to debug and modify. It shows, at a glance, exactly which line of code creates each line of your report, saving hours of needless search time when you must make a change in the report format (and don't you always have to, sooner or later?).

Now you've done your flowcharting and planned the printed reports, you're ready to write the cleanest code you've ever created. And Stirling/Bekdorf's 78C1 Combination Coding/CRT Layouts will help you do it faster. If you're using separate coding forms and CRT layouts now, you'll really enjoy the improvements we've designed for you in the combination 78C1.

Every sheet gives you 2 form uses for the price of one. When display layout and line folding points aren't critical, use 78C1's full 28 line x 80 column grid area for regular program steps. Then for interactive or instructional sections, to plan display layouts simply keep your characters within the appropriate CRT indicator lines, and you'll automatically know where every character will show on your CRT screen.

Gives you 4 extra coding lines on every sheet. Full 28 line x 80 column coding capacity saves you 14 sheets out of every 100, compared to 24-line forms. 86 sheets hold more program steps than 100 sheets of any 24-line form, yet we offer full-size 6mm x 3mm grid blocks to give you comfortable writing room and visual space between lines.

Works with your CRT display, no matter what brand you own. Equipped for both 16 line x 64 column and 24 line x 80 column display formats, 78C1 is compatible with your video terminal.

Available in three separate versions (one for BASIC languages, one for FORTRAN, another for COBOL), the 78C1 is so powerful we include a 7-page instruction manual with every order, to help you discover every facet of its unmatched capabilities.

Every tool in the Stirling/Bekdorf system is surface-engineered to take both pen & pencil without blotching. Our tough, extra-heavy, 22# paper is pure enough to use with critical magnetic ink character readers, although you'll probably never use it that way. We use it because it gives crisp, sharp characters whether you write with pencil or plastic-tip pens.

Every part of our system uses eye-comfortable soft blue grids. All grid rulings, tints, and division rules are reproduced in a special shade of blue. We chose it because it's easy on your eyes, even after hours of continuous programming. If you write software as a professional programmer, you'll particularly appreciate the improvement over the green lines you've been writing on.

A 3-ring binder is one more of our secrets for your success. Every tool in the S/B system is designed to work together in a standard 3-ring binder. All your notes, logic concepts, flowcharts, code, CRT layouts, print-out designs, and documentation can be kept together, in order, in one place. This saves time every time you sit down to write. It also makes debugging and modification immensely simpler. By ensuring everything you create stays together, you assure you can find the area you need to modify when the time comes, whether it's next week or next year. You can readily recapture your original train of thought by scanning your flowcharts, then go directly to the area of code you need to alter. In fact, when you use our system to its fullest, even another programmer can quickly comprehend your original thinking, so he can help you make changes or extensions.

Order your supply of the world's most advanced software development tools, right now, before you hatch even one more bug. Ask your local computer store for Stirling/Bekdorf 78F2's, 78P4's, and 78C1's, or use the coupon below to get your supply.

YES! Please rush the programming aids indicated below:

78F2 Flowchartrix
☐ 2 pads of 50 = $7.90 + $2.85 pkng & shpg
☐ 10 pads of 50 = $34.35 + $6.45 pkng & shpg

78P4 Print-Out Design Sheets
☐ 1 pad of 50 = $7.45 + $3.15 pkng & shpg
☐ 5 pads of 50 = $32.10 + $6.75 pkng & shpg

3-hole punched vinyl pockets for 78P4 Design Sheets
☐ set of 5 for $2.65 + $1.35 pkng & shpg
☐ set of 12 for $5.00 + $1.85 pkng & shpg

78C1 Combination Coding/CRT Layouts
All are the same price, but please check which language format you want:
☐ for BASIC ☐ for FORTRAN ☐ for COBOL

☐ 2 pads of 50 = $6.35 + $1.95 pkng & shpg
☐ 10 pads of 50 = $26.85 + $3.35 pkng & shpg

Texas residents please add 5.5% sales tax to base price. We ship UPS so P.O. Box address must give phone number. Prices shown for packing & shipping are U.S.A. only. International shipping rates are higher, write for details.

☐ Sample of one sheet of each form + associated labels & literature: $2.50

Charge to: ☐ Master Charge ☐ Visa exp. date ____
Card # ____
Enclosed is my check for $____
Signature ____
Name ____
Address ____ Phone ____
City ____ State ____ Zip ____

Stirling/Bekdorf™

4407 Parkwood ☐ San Antonio, TX 78218 ☐ (512) 824-5643

b1

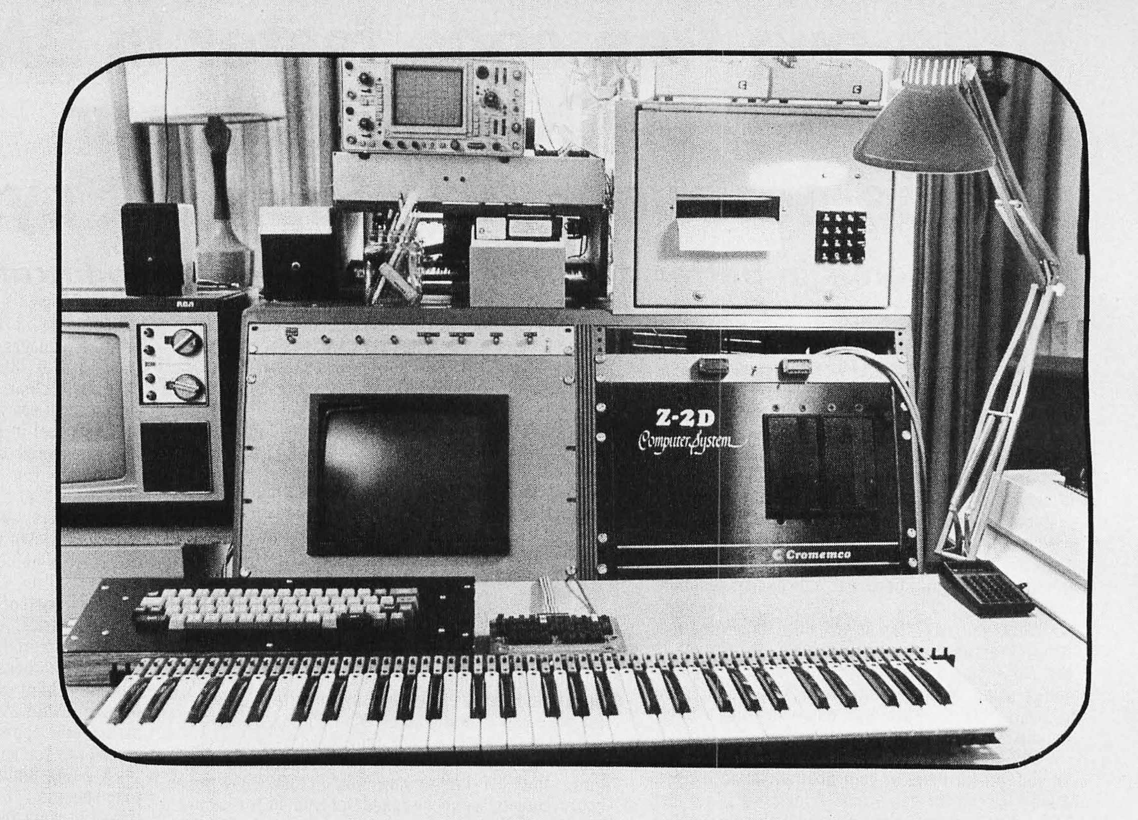


Photo 1: The author's polyphonic music keyboard system, which allows more than one note to be played simultaneously. The scanning keyboard interface is just behind the 61 note manual in the foreground. The stand alone ASCII keyboard in front is dedicated to music related tasks, and, in conjunction with a video terminal (not shown), allows two easily distinguished levels of system control.

Polyphony Made Easy

Steven K Roberts
129 N Galt Av
Louisville KY 40206

Photos by Douglas Fowley

It was not long after the successful implementation of a hardware chromatic tone and envelope generator for my system that I began to wish for a method of playing music that would be somewhat less cumbersome than tune encoding with the ASCII keyboard. The ability to store a melody by defining all the notes and then allowing the computer to perform it was worthwhile, but without some technique for spontaneous interaction the system could hardly be called an instrument. A music keyboard was clearly called for.

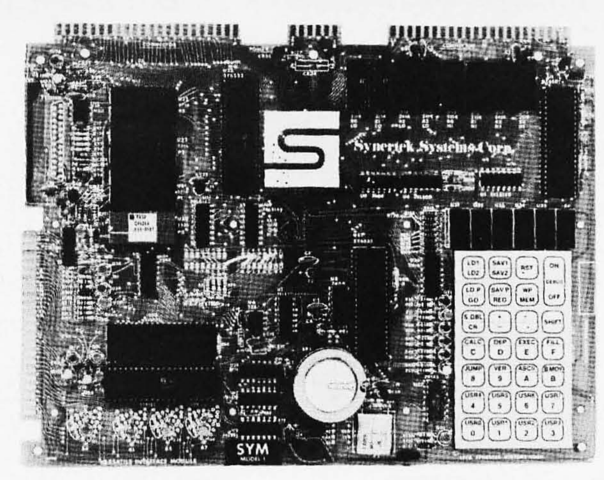
I obtained a 61 note (5 octave) organ manual from the Kimball Organ Company for about $75, and considered the interface task in depth. Among the primary performance specifications for the design were:

- Polyphonic capability (not limited to single notes)
- Undiscernible response delay
- Very low processor overhead

The last of these requirements precluded the use of a software scan, which would have reduced hardware to its simplest form, and the need for polyphony called for either a bit map interface or a multiplexer with its own memory. To simplify the software as much as possible, the latter approach was selected.

A note should be inserted here about the touchy subject of software simplification. An argument frequently heard in the world of the microprocessor is that *everything* should be done with the program. If the processor in question is a dedicated controller, then by all means, all the work that can be reasonably handled by the program should be so assigned. However if the processor is at a higher level of system abstraction, it may be more efficient to delegate certain repetitive tasks either to hardware or to another microcomputer. There is no convenient generalization defining the tradeoff, but in cases where software complexity gets out of hand due to the presence of a fairly mundane but demanding task, some parallelism is usually called for. In the specific

SYM-1, 6502-BASED MICROCOMPUTER



- FULLY-ASSEMBLED AND COMPLETELY INTEGRATED SYSTEM that's ready-to-use
- ALL LSI IC'S ARE IN SOCKETS
- 28 DOUBLE-FUNCTION KEYPAD INCLUDING UP TO 24 "SPECIAL" FUNCTIONS
- EASY-TO-VIEW 6-DIGIT HEX LED DISPLAY
- KIM-1* HARDWARE COMPATIBILITY
  The powerful 6502 8-Bit MICROPROCESSOR whose advanced architectural features have made it one of the largest selling "micros" on the market today.
- THREE ON-BOARD PROGRAMMABLE INTERVAL TIMERS available to the user, expandable to five on-board.
- 4K BYTE ROM RESIDENT MONITOR and Operating Programs.
- Single 5 Volt power supply is all that is required.
- 1K BYTES OF 2114 STATIC RAM onboard with sockets provided for immediate expansion to 4K bytes onboard, with total memory expansion to 65, 536 bytes.
- USER PROM/ROM: The system is equipped with 3 PROM/ROM expansion sockets for 2316/2332 ROMs or 2716 EPROMs
- ENHANCED SOFTWARE with simplified user interface
- STANDARD INTERFACES INCLUDE:
  —Audio Cassette Recorder Interface with Remote Control (Two modes: 135 Baud KIM-1* compatible, Hi-Speed 1500 Baud)
  —Full duplex 20mA Teletype Interface
  —System Expansion Bus Interface
  —TV Controller Board Interface
  —CRT Compatible Interface (RS-232)
- APPLICATION PORT: 15 Bi-directional TTL Lines for user applications with expansion capability for added lines
- EXPANSION PORT FOR ADD-ON MODULES (51 I/O Lines included in the basic system)
- SEPARATE POWER SUPPLY connector for easy disconnect of the d-c power
- AUDIBLE RESPONSE KEYPAD

Synertek has enhanced KIM-1* software as well as the hardware. The software has simplified the user interface. The basic SYM-1 system is programmed in machine language. Monitor status is easily accessible, and the monitor gives the keypad user the same full functional capability of the TTY user. The SYM-1 has everything the KIM-1* has to offer, plus so much more that we cannot begin to tell you here. So, if you want to know more, the SYM-1 User Manual is available, separately.

SYM-1 Complete w/manuals	**$269.00**
SYM-1 User Manual Only	**7.00**
SYM-1 Expansion Kit	**75.00**

Expansion includes 3K of 2114 RAM chips and 1-6522 I/O chip.

SYM-1 Manuals: The well organized documentation package is complete and easy-to-understand.

SYM-1 CAN GROW AS YOU GROW. Its the system to BUILD-ON. Expansion features that are soon to be offered:

8K Basic ROM	**$159.00**
TV Interface Board	**349.00**

QUALITY EXPANSION BOARDS DESIGNED SPECIFICALLY FOR KIM-1, SYM-1 & AIM 65

These boards are set up for use with a regulated power supply such as the one below, but, provisions have been made so that you can add onboard regulators for use with an unregulated power supply. But, because of unreliability, we do not recommend the use of onboard regulators. All I.C.'s are socketed for ease of maintenance. *All boards carry full 90-day warranty.*

All products that we manufacture are designed to meet or exceed industrial standards. All components are first qualtiy and meet full manufacturer's specifications. All this and an extended burn-in is done to reduce the normal percentage of field failures by up to 75%. To you, this means the chance of inconvenience and lost time due to a failure is very rare; but, if it should happen, we guarantee a turn-around time of less than forty-eight hours for repair.

Our money back guarantee: If, for any reason you wish to return any board that you have purchased directly from us within ten (10) days after receipt, complete, in original condition, and in original shipping carton; we will give you a complete credit or refund less a $10.00 restocking charge per board.

VAK-1 8-SLOT MOTHERBOARD

This motherboard uses the KIM-4* bus structure. It provides eight (8) expansion board sockets with rigid card cage. Separate jacks for audio cassette, TTY and power supply are provided. Fully buffered bus.

VAK-1 Motherboard **$129.00**

VAK-2/4 16K STATIC RAM BOARD

This board using 2114 RAMs is configured in two (2) separately addressable 8K blocks with individual write-protect switches.

VAK-2 16K RAM Board with only 8K of RAM (½ populated)	**$239.00**
VAK-3 Complete set of chips to expand above board to 16K	**$175.00**
VAK-4 Fully populated 16K RAM	**$379.00**

VAK-5 2708 EPROM PROGRAMMER

This board requires a +5 VDC and ±12 VDC, but has a DC to DC multiplyer so there is no need for an additional power supply. All software is resident in on-board ROM, and has a zero-insertion socket.

VAK-5 2708 EPROM Programmer **$269.00**

VAK-6 EPROM BOARD

This board will hold 8K of 2708 or 2758, or 16K of 2716 or 2516 EPROMs. *EPROMs not included.*

VAK-6 EPROM Board **$129.00**

VAK-7 COMPLETE FLOPPY-DISK SYSTEM (Feb. '79)

VAK-8 PROTYPING BOARD

This board allows you to create your own interfaces to plug into the motherboard. Etched circuitry is provided for regulators, address and data bus drivers; with a large area for either wire-wrapped or soldered IC circuitry.

VAK-8 Protyping Board **$49.00**

POWER SUPPLIES

ALL POWER SUPPLIES are totally enclosed with grounded enclosures for safety, AC power cord, and carry a full 2-year warranty.

FULL SYSTEM POWER SUPPLY

This power supply will handle a microcomputer and up to 65K of our VAK-4 RAM. ADDITIONAL FEATURES ARE: Over voltage Protection on 5 volts, fused, AC on/off switch. Equivalent to units selling for $225.00 or more.

Provides +5 VDC @ 10 Amps & +12 VDC @ 1 Amp
VAK-EPS Power Supply **$125.00**

KIM-1* Custom P.S. provides 5 VDC @ 1.2 Amps and +12 VDC @ .1 Amps
KCP-1 Power Supply **$41.50**

SYM-1 Custom P.S. provides 5 VDC @ 1.4 Amps
VCP-1 Power Supply **$41.50**

*KIM is a product of MOS Technology



ENTERPRISES
INCORPORATED

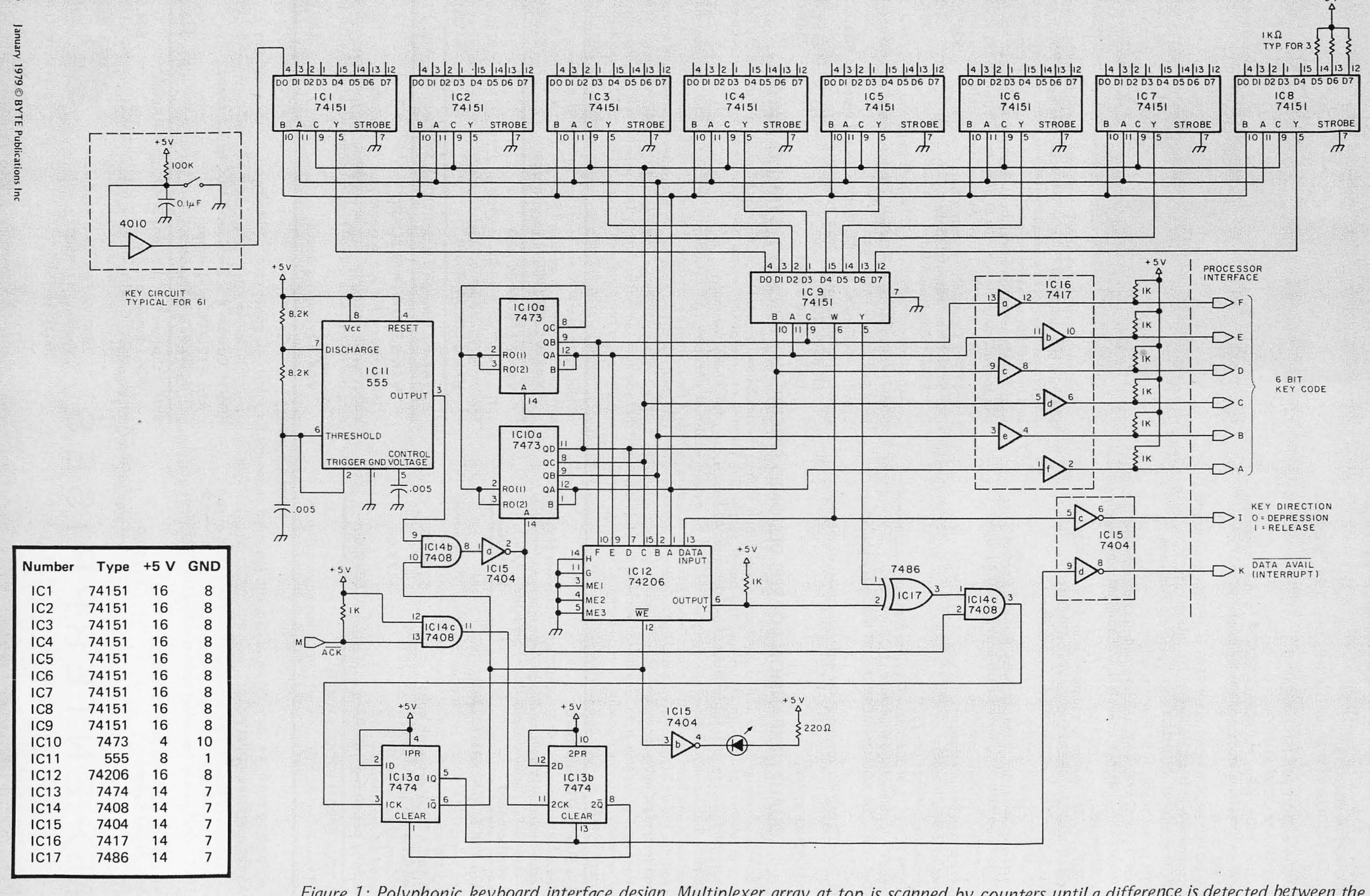
2967 W. Fairmount Avenue
Phoenix AZ. 85017
(602)265-7564





Prices in effect Nov. '78

Circle 318 on inquiry card.

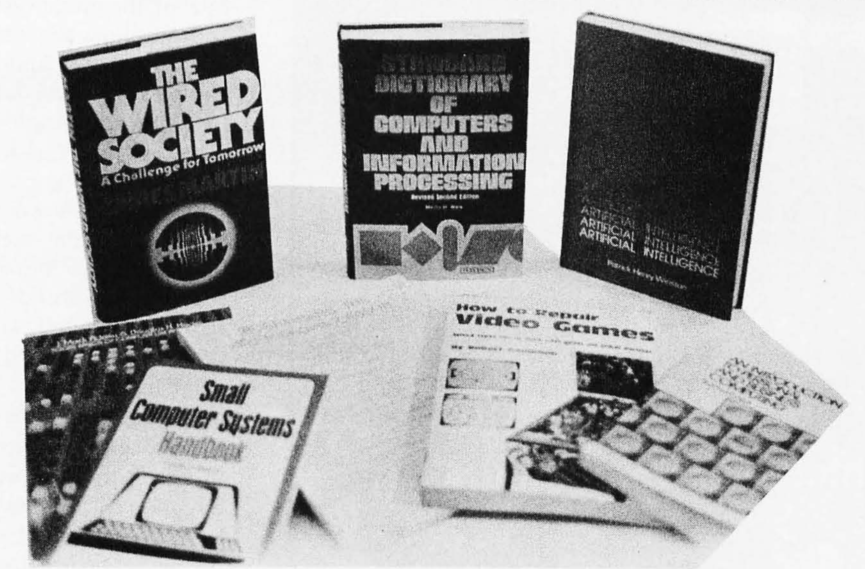


Number	Type	+5 V	GND
IC1	74151	16	8
IC2	74151	16	8
IC3	74151	16	8
IC4	74151	16	8
IC5	74151	16	8
IC6	74151	16	8
IC7	74151	16	8
IC8	74151	16	8
IC9	74151	16	8
IC10	7473	4	10
IC11	555	8	1
IC12	74206	16	8
IC13	7474	14	7
IC14	7408	14	7
IC15	7404	14	7
IC16	7417	14	7
IC17	7486	14	7

Table 1: Power wiring table for figure 1.

Figure 1: Polyphonic keyboard interface design. Multiplexer array at top is scanned by counters until a difference is detected between the addressed key and the corresponding bit in memory. The scan is stopped and the key address and direction are read by the processor, allowing the scan to continue.

An Assortment of Solutions



for your Resolutions

ARTIFICIAL INTELLIGENCE by Patrick Henry Winston

☐ This book is designed for use in a course on artificial intelligence, but should prove invaluable to the newcomer to the topic, as well as being useful to the experienced as a reference text. Part One is set up to be used as an introduction to the key ideas of the subject, including the basics of elementary representation, symbolic constraint exploitation, search, and control. In addition, understanding natural language, representing common sense meaning, computer vision, and expert problem solving are covered. Part Two covers the programming language LISP, its organization and uses. Profusely illustrated. 444 pp. Only $10.00 hardcover.

HOW TO REPAIR VIDEO GAMES by Robert Goodman

☐ A 270-page service/repair manual containing information on products sold by manufacturers of electronic home video games and pinball machines. Each chapter of the manual, is devoted to an individual manufacturer's equipment. Some manufacturers covered are equipment manufacturers while others produce game chips only. The three game chip manufacturers are General Instrument, Texas Instrument and National Semiconductor, while equipment manufacturers include Magnavox, Atari, Radio Shack, RCA and Midway. $7.95.

INTRODUCTION TO PERSONAL AND BUSINESS COMPUTING by Rodnay Zaks

☐ This text is designed for both the novice and the user seeking to expand his/her knowledge in the microcomputer field. A comprehensive introduction to concepts, peripherals, and techniques, this book serves well as an educational text. 245 pp. Only $6.95.

PROGRAMMING THE 6800 MICROPROCESSOR by Bob Southern

☐ This self-instruction workbook is a guide to the fundamentals of assembly language and machine code programming of the 6800 microprocessor and its peripheral devices. Considerable coverage is given to programming of input and output devices. The asynchronous communications interface adapter, each with their various modes of operations, are explored in detail in both noninterrupt and interrupt modes. Program design and documentation are emphazied along with programming hints and aids. 270 pp. $12.95.

SMALL COMPUTER SYSTEMS HANDBOOK by Sol Libes

☐ This is an introductory text for those with a minimal knowledge of electronics who are interested in microcomputers as a hobby. The author explains microcomputer concepts from number systems and logic circuits to programming languages and applications with the emphasis on practical knowledge. 197 pp. $8.45.

SOCIAL EFFECTS OF COMPUTER USE AND MISUSE by Adams/Hayden

☐ Most of us have come to accept the presence of computers in our lives. Their effect on society as a whole is worthy of careful consideration. This book surveys the evolution of computers, their application, and the social, economic and philosophical implications of their increasing use. 326 pp. $10.00.

STANDARD DICTIONARY OF COMPUTERS AND INFORMATION PROCESSING by Martin H Weik

☐ This is a very complete, fully cross-referenced dictionary. It goes a step farther in that it includes full explanations, practical examples, many pertinent illustrations, and supplementary information for over 12,500 hardware and software terms. It cross-references the terms to other closely related concepts, and appended to each definition, as the need arises, are explanations, tutorial information, examples, usage areas, and cross-references for further clarification of concepts and meanings. 390 pp. $16.95 hardcover.

WIRED SOCIETY by James Martin

☐ This is a fascinating and not-too distant account of what the future might be like, changed by the new telecommunications technology. Prefacing his discussions with historical as well as contemporary references, Martin lures the reader into an impending technical revolution, which will change corporations, change government, and changes the lives of everybody. $12.95 hardcover.

For your convenience in ordering, please use this page plus the order form on page 99. You may photocopy this page.

DIAL YOUR CHARGE CARD ORDERS TOLL FREE (800) 258-5477.

master charge — VISA

BITS™ inc Books to erase the impossible

POB 428, 25 Route 101 West
Peterborough, NH 03458

Circle 35 on inquiry card.

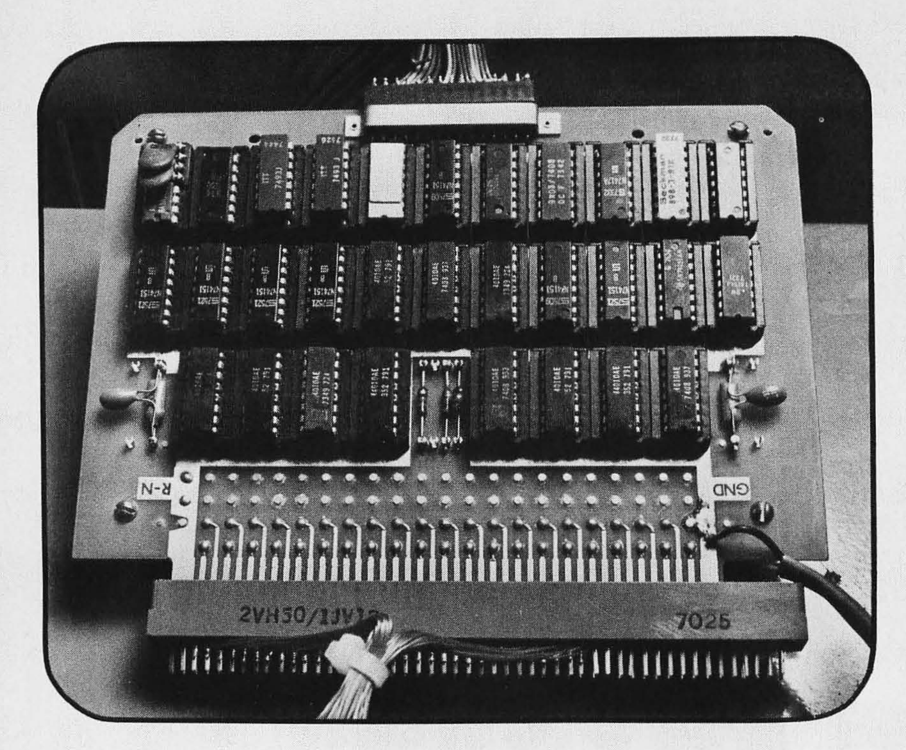


Photo 2: Keyboard interface hardware. The circuit fits perfectly onto a Robinson-Nugent 30 socket wire wrap panel. It could just as easily be implemented on an S-100 card if there is no objection to the wire bundle. Any combination of simultaneous key depressions and releases on the musical keyboards, at any practical speed, will result in a series of asynchronous "change of state" notices to the processor, which remains ignorant of interface function at all other times. The software maintains a list of keys currently depressed and deals with them appropriately. A simple handshaking scheme makes the interface synchronous with the host computer.

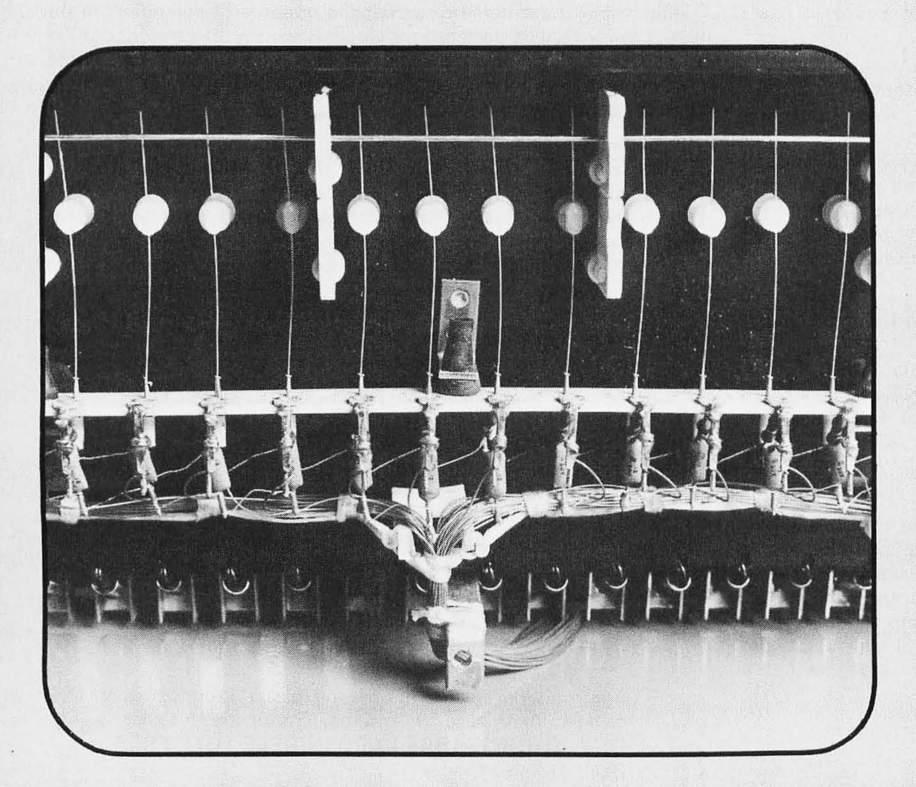


Photo 3: Underside of organ manual. The resistor-capacitor networks shown are used with each key for noise reduction. The wire bundle exiting at the bottom terminates in the interface's edge connector.

case of the music system, keyboard scanning and envelope generation can impose such a burden that the ability to *simultaneously do* complex real time data manipulation is lost.

Thus, optimization of the keyboard interface design was undertaken with the system considerations given uppermost priority, and the result is shown in figure 1. The 61 key switches of the manual are scanned completely every 5 ms, and with each step of the scan the position of the presently addressed key is compared with its last known position, which is stored in a 256 by 1 memory segment. If there is no difference, the scan proceeds, but if the key has changed state, the processor is interrupted with the binary value of the key in question along with a direction bit, and the corresponding memory location is changed to reflect the new status of the keyboard. The scan is suspended until the information is accepted by the computer. In this fashion, any combination of simultaneous key depressions and releases, at any practical speed, will result in a series of asynchronous "change of state" notices to the processor, which remains ignorant of interface function at all other times. The software maintains a list of keys currently depressed, and deals with them appropriately.

Action of the scanning interface is synchronized with the host computer by means of a simple handshaking scheme: when a change of state is detected, the data available signal appears (this may be treated as an interrupt or polled periodically, depending upon available time). It is then the processor's job to read the input port upon which the 6 bit key code and the direction bit appear, whereupon the acknowledge signal is created by the port strobe, allowing the scan to continue. It is important to note that the scan stops whenever a change is encountered (awaiting processor intervention) because ultimate keyboard servicing time is then largely a function of the support software. In the unlikely event that a user of this system chooses to implement such real time functions in BASIC, it will be found that a forearm laid in jest upon the keyboard results in a sweep up the musical scale lasting on the order of a second. In a more realistic situation (assembler level coding) the delay is unnoticeable.

Construction, of course, should follow the usual procedures required of random logic interfacing. Cables between the board and the processor should be kept short, with intervening grounds between the handshake lines. The keyboard shown in photo 1 consists of simple normally-open contacts; their noise is filtered by the resistor-capacitor (RC)

networks shown in the schematic. Any remaining bounce may be trimmed out by adjusting the clock rate, which is nominally 12.5 kHz. There is one light emitting diode (LED) on the board to provide a visual check of operation: "Key In Progress" is lit between the data available signal and the acknowledgment signal.

The circuit concept is directly expandable to accommodate many more inputs, with the memory and counter capable of addressing up to 256 points. This interface concept would be quite at home in many industrial control environments, as well as anywhere a large number of contact closures must be observed.

In the music system shown in photo 1, the keyboard interface has provided the much needed flexibility in the interactive utilization of the instrument. The "feel" is not unlike that of a standard electronic organ, and with the available processing horsepower taken into consideration, the unit is a composer's delight. From teaching applications (where the human must correctly repeat "by ear" a computer generated phrase) to the support of creative effort, an efficiently integrated polyphonic keyboard is an essential link between artist and computer. ■

Grandmaster Walter Browne versus Chess 4.6

John R Douglas
25 Cannon Dr
Elk River MN 55330

About the Author

J R Douglas has had a professional interest in computers since 1959. For the past ten years, he has been a designer of "larger and larger" computers for Control Data Corp. He and David Cahlander (also with CDC) act as operators of the data communications system on those occasions when Chess 4.6 plays in competition. They also assist in interfacing the Chess program to the operating system and in maintaining the opening book data base.

The organizers of Minnesota's Twin Cities Open invited Northwestern's Chess 4.6 and Control Data's CYBER 176 to take part. Since this World Champion computer system had not been tested in open human competition, its programmers readily accepted the invitation. With a rating of 1936, Chess 4.6 was seeded number one in the 5 round Swiss system tournament, perhaps the first time a computer has been accorded that position.

A new electronic chessboard was used for the first time. The microprocessor which controls the board senses the opponent's moves magnetically, transmits the move in algebriac notation via telephone to CYBER 176, and then indicates CYBER 176's responses by illuminating small lights on the square of the piece to be moved and on the one to which it is to go. Chesstor, as this device is called, also senses the hitting of the chess clock and thus keeps track of the time used.

Chess 4.6 did very well in this event, wrapping up first place in four rounds and finishing the tournament with a perfect 5–0 result. The 30 to 35 points gained brought its rating close to the elusive Expert barrier. This win carried with it an invitation to enter US Champion Walter Browne's simultaneous exhibition.

A large crowd gathered to listen to Grandmaster Browne's lecture and to watch his 44 board simultaneous exhibition. Asked his opinion of computer chess, Browne replied that he did not mind their slow style of play. Browne invited wagers on his computer game and was surprised to find takers in the audience. Later he declined the side bets, saying that although he would win, the time he spent at the computer's board would not be fair to the other players.

Grandmaster Browne's 1978 tour had thus far produced an amazing winning streak, with only two losses and six draws in 17 exhibitions. As this exhibition began, everyone was amazed at Browne's pace around the first six circuits; he barely paused at each board before responding to the position before him. He certainly did not seem concerned about the computer's Benoni opening, which requires sharp and exact play. (Later Browne acknowledged that he should have spent more time with this opening.) Browne did, however, appear to be perplexed by the computer's advantage out of the opening, and this put an end to his making blitz mode moves at the computer's table. Here is the way the game went.

Benoni Defense

White: W Browne Black: Chess 4.6

1 P-Q4 N-KB3 2 P-QB4 P-B4 3 N-KB3 PxP 4 NxP P-K4

Chess 4.6 tends toward sharp opening play.

5 N-N5

Walter Browne tends toward sharp opening play.

5 . . . B-B4

Chess 4.6 has been modified for this event to allow the operator to wait until the visiting Grandmaster arrives before requesting the computer's move. This gives the system maximum time for computation.

6 QN-B3 Castles

Browne's Knight move brings 4.6 out of the opening book. At this point Chess 4.6 has used two minutes and Walter Browne has hardly broken stride as he passed. But now the skid marks in front of the electronic chessboard are added to each time Browne passes by.

7 P-K3 P-Q3 8 B-K2 P-QR3

The chessboard seems to flicker with electronic pride as the machine's estimate of its advantage climbs to more than one half pawn — more than one half a Grandmaster's pawn.

9 N-R3 N-B3

Browne's charming wife spent a lot of time watching the computer and was dismayed as with each successive move the computer's evaluation routine gave a stronger and stronger procomputer assessment of the position.

10 N-B2 B-B4

Chess 4.6 is predicting Browne's moves with impressive accuracy, and it gets about four minutes per move in which to thwart Browne's design for the game.

11 0-0

4.6 expected N-Q5

11 . . . Q-Q2

Still retaining a one half pawn lead and expecting 12 B-Q3 P-KR4 13 N-Q5 N-KN5 14 P-K4.

12 P-QN3 K-R1 13 B-N2 R-KN1

4.6 is having trouble finding something to improve the position. It predicts 14 Q-Q2 P-R3 15 QR-Q1 N-K5 16 NxN BxN.

14 N-R4 B-R2

4.6's position seems solid now.

15 B-R3 P-R3 16 R-B1

[*Not 16 BxP? BxN . . .* Burt Hochberg]

16 . . . QR-Q1 (expecting Q-Q2) 17 N-N4 NxN 18 BxN

All anticipated by 4.6; at this point the 4.6 corner fell prey to a bit of mild panic and requested the next move about three minutes before Browne was due back at the board. The new electronic chessboard seemed not to have sensed the previous move (which indeed it had), and in the flurry of the operations following, Chess 4.6 committed its only error.

18 . . . Q-B2

There may be a stronger move.

The program Chess 4.6 was written by David Slate and Larry Atkin of Northwestern University. The *Chesstor* electronic chessboard was constructed by David Cahlander of Control Data Corp.

GLOSSARY

Algebraic notation: a system of recording chess moves characterized by the assignment of the letters a thru h to the files (columns of squares) and the numbers 1 thru 8 to the ranks (rows of squares) done from the point of view of the player with the White pieces. Usually certain abbreviations are employed. The World Chess Federation (FIDE) strongly endorses the use of algebraic notation (see "descriptive notation").

Annotations of ! or ?: represent someone's opinion of the quality of the move. The exclamation point shows a good move; the question mark shows a poor move.

Benoni opening: the name given to the particular set of opening moves which are played in this game. It is a forceful defense by Black which often leads to great tactical display. It was first studied by Reinganun of Frankfort in 1825, and was a favorite defense of two World Chess Champions, Alekhine and Tal.

Blitz move: a move made very quickly.

Chess clock: a device used to put a time limit on a chess game. It is a mechanism with two clock movements and faces, one for each player. When a player makes a move, he or she presses one of two buttons on the top of the clock which starts the opponent's clock movement and stops his or her own. In tournaments a player must make a prescribed number of moves in a limited timespan.

Descriptive notation: the traditional system of recording the moves of a chess game, used in this article. A file is named after the piece positioned on it at the start of the game. The ranks are denoted numerically from the point of view of the player having the move.

Expert class: the class of chess player who has a tournament rating of somewhat over 2000 points. As a comparison, the average rating for all members of the United States Chess Federation is around 1300.

Kingside attack: an aggression on the side of the board on which the Kings were positioned at the start of the game.

Pawn up: to have a material advantage of one pawn or its equivalent, or to have a positional advantage of the same value. Under most circumstances, the advantage of one pawn is sufficient for victory.

Piece up: similar to, but better than, being a pawn up; usually leading to certain victory.

Ratings: a guide to the ability of a chessplayer, based on historical data of performance in over-the-board competition in rated chess tournaments. The system most commonly in use was invented by Dr Arpad Elo. A player gains rating points by winning a tournament game, and loses points by losing a game. The effect of a draw on a rating depends on the difference in rating between the two players in the drawn game. A lower rated player achieving a draw with a higher rated player will gain points, while the higher rated player will in all probability lose points.

Sharp play: playing moves which are likely to lead to great tactical display.

Simultaneous exhibition: an event in which a very strong chess player engages in competition against many weaker opponents in many separate games.

Swiss system tournament: a popular type of chess tournament in the United States. It is usually conducted as follows: in the first round, the players are ordered according to their ratings. Then the top player in the upper half is paired against the top player in the lower half, and so on down to the bottom player in each half. For each game, each player is given one point for a win, one half point for a draw, and zero for a loss. In the second and later rounds players are paired according to the following ordered general principles:

1. A player must not be paired with any other player more than once.
2. Players with equal scores after each round must be paired as much as possible.
3. Colors are assigned by the director of the tournament as equitably as possible.

19 Q-K1

Now 4.6 is only one quarter pawn up, but it thinks 19 Q-Q2 would have been more to the point.

19 . . . B-B4 20 B-KB3 B-Q6

One half pawn up again. Computer fans could relax a bit. Both 4.6 and Browne see the next four moves now. Browne spends a lot of time at our table. Mrs Browne spends a lot of time warning Walter that he should spend a lot of time at our table. She asked how much the machine cost and was told several million dollars. She told Walter that the machine said it had the advantage. Browne was not impressed, although he did start to thump the pieces and our clock as he passed. After a lengthy stay at the board he played.

21 BxB PxB 22 B-K2 B-B4 23 P-B3 P-K5 24 P-B4 B-Q2 25 N-B3 Q-R4

Now the visting master stops, does a double step, smiles: he's got this thing now. The Queen is out of play. Browne begins a Kingside attack, smiles at the spectators, savors his move.

26 Q-R4

Thump, smile. It's lucky that the electronic board can only sense the position of the piece and not the force with which it is moved, for Browne's forceful play intimidates the spectators.

26 . . . B-B3

For the next four moves, 4.6 must find defensive resources that are not obvious to those in attendance, and predictions of an early end to the game begin.

27 R-QB2

4.6 expected P-KN4.

27 . . . P-QN4

Can Browne be distracted?

28 P-KN4

Doesn't look like he is.

28 . . . P-N5 29 N-Q1 R-Q3

Now everyone sees the defense. 4.6 expects 30 P-N5 N-R2 31 N-B2 KR-Q1 32 B-N4 Q-N3 33 B-B5. Browne plays.

30 N-B2 R/1-Q1

If 4.6 survives, the Grandmaster is not going to like what's happening on the Queen file.

31 R-Q1 RxRch 32 BxR

Small thump, walk away, stop, look back, frown.

32 . . . R-Q3

At this point the clocks show two hours 44 minutes for 4.6 and 22 minutes for Grandmaster Browne.

33 Q-N3 Q-Q1

4.6 correctly projects Browne's game for the next 11 moves.

34 R-B1

Now Browne is defending. Things are not going well at some of the other boards, either, but it is here that Browne spends most of his time.

34 . . . R-Q7

What a nice place for one's Rook.

35 P-N5

4.6 had been expecting this much earlier.

35 . . . PxP 36 PxP N-R2 37 P-N6 PxP

We are a whole pawn up!

38 QxP Q-R5!

Now things start to liven up. Browne looks unnerved and spends a long time on his next move. He gets into trouble at the next board, too. Here he finds:

39 Q-B5 B-Q2

4.6's backers find happiness and will admit that they just may have moved the piece more slowly and punched the clock more vigorously than was necessary, so much so that Browne remarked, with just a hint of a smile, "It is not allowed for the computer

1.	d4	Nf6	33.	Qg3	Qd8
2.	c4	c5	34.	Rc1	Rd2
3.	Nf3	c:d	35.	g5	h:g
4.	N:d4	e5	36.	f:g	Nh7
5.	Nb5	Bc5	37.	g6	f:g
6.	N/1c3	0-0	38.	Q:g6	Qh4
7.	e3	d6	39.	Qf5	Bd7
8.	Be2	a6	40.	Qf4	Q:f4
9.	Na3	Nc6	41.	e:f	e3
10.	Nc2	Bf5	42.	Ne4	e2
11.	0-0	Qd7	43.	B:e2	R:e2
12.	b3	Kh8	44.	N:c5	Bc8
13.	Bb2	Rg8	45.	Rd1	Re8
14.	Na4	Ba7	46.	a3	b:a
15.	Ba3	h6	47.	Ra1	g5
16.	Rc1	R/a d8	48.	f:g	Re5
17.	Nb4	N:b4	49.	b4	a5
18.	B:b4	Qc7	50.	Nd3	R:g5+
19.	Qe1	Bc5	51.	Kf2	a:b
20.	Bf3	Bd3	52.	N:b4	Ra5
21.	B:c5	d:c	53.	Ke3	Be6
22.	Be2	Bf5	54.	Kd4	Ng5
23.	f3	e4	55.	Nc2	a2
24.	f4	Bd7	56.	Nb4	Ra4
25.	Nc3	Qa5	57.	Kc5	Ne4+
26.	Qh4	Bc6	58.	Kb5	Bd7+
27.	Rc2	b5	59.	Nc6	Nc3+
28.	g4	b4	60.	Kc5	B:c6
29.	Nd1	Rd6	61.	K:c6	R:c4+
30.	Nf2	R/g d8	62.	Kd6	Rd4+
31.	Rd1	R:d1+	63.	Ke5	Rd1
32.	B:d1	Rd6	64.	resigns	

Table 1: The score (record of moves) of the simultaneous exhibition game between Grandmaster Walter Browne and the computer program Chess 4.6, duplicated here in convenient table form. The notation here is algebraic, as opposed to the descriptive notation which is used in the article text. In this type of notation, the colon (:) indicates a capture and the plus sign (+) indicates check. Moves given as square designations alone are pawn moves.

Circle 202 on inquiry card.

to play psychologically." 4.6 correctly projects Browne's next seven moves.

40 Q-B4 QxQ 41 PxQ P-K6

World Champion 4.6 admonishes US Champion Browne to be careful. Grandmaster Browne admonishes 4.6 not to be overconfident.

42 N-K4 P-K7!

Forcing the exchange of a Bishop for two pawns.

43 BxP RxB 44 NxP B-B1 45 R-Q1 R-K1

Browne now finds a move to make the endgame playable. 4.6 did not anticipate this continuation.

United States Chess Federation
Suggested Rules of Play Involving Computational Machinery

The following rules are suggested for use in USCF rated tournaments when one or both players is a computer. In matters not covered by these rules, play is governed by the FIDE (Fédération Internationale des Échecs) Laws, by FIDE Interpretations of the Laws, and by the USCF Tournament Rules and Pairing Rules, interpreted by the arbiter. In such games the player shall be considered to be the chess algorithm being executed on a specific computer.

The following regulations shall govern play:

1. For the algorithmic player (computer), a piece shall be deemed "touched" when a move involving that piece is communicated.
2. A move shall be deemed executed when the move has been executed on the playing chessboard. Only after this shall the opponent's clock be started.
3. The computer and/or the operator shall keep the score of the game.
4. If, during a game, different positions should arise on the playing chessboard and on the chessboard or representation thereof maintained by the algorithmic player, such differences shall be corrected with the assistance of the arbiter by consulting both players' game scores. In resolving such differences, the player whose score has the correct move, but who has executed a wrong one, has to accept certain disadvantages.
5. If, when such discrepancies occur, the game scores are also found to differ, the moves shall be retraced up to the point where the scores agree, and the arbiter shall readjust the clocks accordingly.
6. The algorithmic player's operator(s) shall have the following duties:
   (a) To make the moves of the algorithmic player on the playing chessboard.
   (b) To communicate the moves of the opponent to the algorithmic player.
   (c) To operate the chess clock for the algorithmic player.
   (d) To inform the algorithmic player, at its request, of the time consumed by either or both players.
   (e) To claim the game in cases where the time limit has been exceeded.
   (f) To carry out the necessary formalities in cases where the game is adjourned.
   (g) To communicate proposals of a draw between the algorithmic player and the opponent.
   (h) To carry out the functions associated with machine communication failure. During restart, program parameters must be reset to the most recent values. Board position and status, along with clock time, may also be entered.
7. The opponent may appoint a deputy to record the game score.
8. Communication to and from the algorithmic player regarding the moves of the game shall be made in a standard (clear and unambiguous) notation.
9. During the course of a game, an algorithmic player may not request additional data or information which requires human intervention. Such a request shall be considered a violation of Article 19.1a of the Laws. *[Article 19.1a says that during a game a chess player may not use any knowledge which is not his/her/its own . . .* **RS***]*
10. With the approval of the arbiter in advance of the first round, the operator may resign or accept a draw on behalf of the algorithmic player.

Source: *Official Rules of Chess,* second edition, edited by Martin E Morrison, David McKay Company Inc, New York, 1978, pages 112 and 113. Reprinted by permission.

46 P-QR3 PxP

Greedy, but what else?

47 R-R1 P-N4 48 PxP R-K4

Expecting N-R4.

49 P-N4 P-R4

It's a whole piece up.

50 N-Q3 RxPch 51 K-B2 PxP 52 NxP R-QR4 53 K-K3 B-K3 54 K-Q4 N-N4

Expecting 55 N-B6 R-KB4 56 N-Q8 R-B5.

55 N-B2

Browne offers a draw. The computer's corner is divided: do we play on for science or bag a draw with a Grandmaster for the record book? Browne returns in less than two minutes. Science prevails: 4.6's Bishop and Knight tactics should be interesting.

55 . . . P-R7 56 N-N4 R-R5

The machine records a 5 point lead for Black.

Browne first plays K-B3, hits the clock and moves away: then he skids to a stop, returns, and announces, "That's not my move," stands, leans, sways, taps, pounds on a loose pawn, and finally plays:

57 K-B5

Browne has missed his last chance at the Rook pawn.

57 . . . N-K5ch 58 K-N5 B-Q2ch 59 N-B6 N-B6ch (the final nail) 60 K-B5 BxN 61 KxB RxPch 62 K-Q6 R-Q5ch 63 K-K5 R-Q8

And Browne resigns.

In the course of this simultaneous exhibition, Chess 4.6 correctly projected 35 of Browne's 58 moves, not including those from the opening library. The largest number of positions examined for any one move was 2,158,456 in just over nine minutes. In total, Chess 4.6 used just over four hours of computation time, whereas Grandmaster Browne spent 26 minutes at the board.

Browne was not pleased with his performance in general; six wins and six draws were scored against him, dropping his percentage for the day to 73%. But the game Chess 4.6 played will certainly provide some pleasant moments to those who play over the moves, and they're almost certain to find the computer's style refreshing.

Editor's Comments

After seeing Douglas' article, our readers are advised against being misled about the strength of play of which the program Chess 4.6 is capable. In this particular game, the computer looks very good. It should be remembered, however, that Grandmaster Browne was playing with a number of disadvantages.

This was a simultaneous exhibition, and therefore Browne was playing 44 games at once, including the computer's game.

Browne had to make rapid judgments about moves in 44 games, whereas Chess 4.6 concerned itself with only one game. A side effect of this difference in workload was that the computer could use at least twice as much time as is normally available to it for the calculation of its moves.

Browne also had the disadvantage of not knowing his opponent's strengths and weaknesses. If the human had chosen to play a more closed and positional game, the computer would have looked far worse. Many chessmasters who play simultaneous exhibitions adopt the following strategy: choose sharp lines of play so that the weak opponents may be quickly vanquished, thereby giving more time for concentrating on the stronger players. Unfortunately for Browne, this method played directly to the computer's strong area.

Since this event, Walter Browne has greatly increased his interest in computer chess. It is rumored that he may play another simultaneous exhibition in which all of the opponents are computer programs. Other chessmasters may also be attracted by this new field. Even though the original Levy wager period is ended, computer chess will continue to be a field of great interest to researchers on the cutting edge of computer technology.... RS■

United States Chess Federation
Procedure for Registering Chess Playing Computer Programs

In order to allow for legitimate scientific testing of chess playing computer programs, but at the same time to protect the accuracy of the rating assigned to the programs and their opponents, the following procedures are used to register chess playing computer programs.

1. No chess playing computer program may be sold a United States Chess Federation (USCF) membership to allow it to participate in a USCF rated tournament.
2. Any new chess playing computer program must apply for special registration at least one month in advance of its first participation in a USCF rated tournament.
3. Normally, the programmer is the person who may register the chess playing computer program.
4. The following information is necessary for registration:
   A. Name and address of programmer.
   B. Name and pertinent identification of the computer (or computers, if more than one may be used to run the program) the program will be using.
   C. A general written description of the program procedure used in playing. A printout of the program is not usually needed.
5. Any significant program or computer changes which take place after the initial registration must be filed at least one month in advance of participation in a rated tournament with the modifications.
6. Requests for registration should be sent to Craig W Ellyson, member of the USCF Ratings Committee, 11 Woodland Dell Rd, Wilbraham MA 01095. The applicant will be notified by the USCF National Office of the disposition of his application, and, if it is approved, he will be asked for the $15 annual registration fee and be issued a certificate permitting the program/computer to participate in USCF rated tournaments.

An Introduction to BNF

W D Maurer
Rm 634
University Library Bldg
George Washington University
Washington DC 20052

BNF is a standardized method of abbreviating certain statements which are made about a programming language when it is being strictly defined, as in a programming manual. As such, BNF bears an analogy to the use of algebra in order to simplify certain statements which are made about physical and mathematical quantities. Thus the statement that the volume of a sphere is equal to four-thirds the cube of the radius times the ratio of the circumference of a circle to its diameter may be abbreviated

$$V = \frac{4\pi r^3}{3}$$

BNF is used like algebra: It simplifies certain statements made about languages in the same way that algebra simplifies certain statements made about mathematical and physical quantities.

In order to make abbreviations such as the one above, we set up various conventions. For example:

1. The quantities in the statement are represented by single letters; thus V stands for the volume.
2. Squares, cubes, and other powers are represented by superscript notation; thus r^3 is the cube of r.
3. Certain fixed quantities which appear very often have standard names; thus the ratio of the circumference of a circle to its diameter is always denoted by π.
4. Two single letters written together signify "times"; thus πr means π times r. (This rule must be amplified in order to specify clearly that πr^3 means π times the cube of r, and not the cube of πr; and to make clear that it also applies to numbers, so that 4π means 4 times π.)

We shall now set up a number of similar conventions in order to abbreviate statements made about programming languages. For example, consider the following sentence:

A GO TO Statement in FORTRAN consists of the words GO TO followed by a statement number.

We may abbreviate this in BNF as follows

< GO TO statement > ::=
'GO TO' < statement number >

Author's Note

Throughout this exposition we shall use the mnemonic, BNF, without specifying what it stands for. Originally BNF was called Backus Normal Form, after John W Backus, who was the head of the project at IBM that developed the first FORTRAN and who proposed the present scheme in connection with the conference that developed ALGOL. Later it was pointed out that BNF is not really a normal form, in the sense in which this phrase is used in mathematical logic. Some people today prefer Backus Naur Form as a name for BNF; this honors Peter Naur, who edited the ALGOL Report [1] in which BNF was first used to define the syntax of ALGOL.

In doing this we have implicitly set up the following conventions:

1. The signs < and >, which also stand for "less than" and "greater than" but in this context are called *angle brackets*, enclose the name of some "quantity" which we wish to define in the programming language. We call such a "quantity" a *syntactical variable*.
2. The special sign ::= means "is defined as." This comes from ALGOL, in which the sign := is the replacement symbol, used in statements such as A:=B (i.e., set A equal to B).
3. The words "followed by" may be omitted, just as "times" may be omitted in algebra.

There is a further analogy between BNF and algebra. When we write 'GO TO' <statement number>, we mean the words GO TO followed by *any* statement number. This is very much like writing 3x to mean 3 times the value of x, whatever it happens to be. Here x is a variable, but 3 is a constant. Similarly, the phrase < statement number > is a syntactical variable, and may stand for any of various statement numbers; but 'GO TO' always stands for the same thing, and may thus be called a *syntactical constant*. Syntactical constants are subject to another rule:

4. A syntactical constant is enclosed in quotes.

Circle 393 on inquiry card.

This last rule, incidentally, is not always followed. The single quote character ' is actually meaningful in some programming languages, and its use in programming language definition would thus cause confusion here. Of course, we can always use the double quote " instead of the single quote, unless the programming language uses both of these symbols (like SNOBOL, for instance). But sometimes even when there is no confusion the quotes are omitted for the sake of brevity.

It is clear, of course, that statements about programming languages may be abbreviated even further. We might write G → 'GO TO' S, thus incorporating the use of single letters for variables, as is done in algebra. In fact, this type of abbreviation is used extensively in the theory of context-free languages. (See references 2 and 3 for two interesting applications of this theory and this type of abbreviation to programming languages.) The trouble with abbreviating this far is that now the abbreviation is not self-contained. We must still make some statement such as "where S stands for a statement number." In contrast, the BNF rules which we define here permit the entire syntax, or "grammar rules" of a language, to be specified in a precise manner, using no other information than that contained in the BNF rules themselves. The semantics, or "meaning" of the language, must still be specified separately; and at this time there is no easy and fairly universal way to specify semantics, although attempts have been made (see references 2 and 4).

Rules in BNF may be extremely simple. We may write

```
< statement number > ::=
        < unsigned integer >
```

to specify that the syntactical variable "statement number" takes the same form as the syntactical variable "unsigned integer." This is often convenient when several syntactical variables have the same form. In most languages, for example, simple variable names, array names, and function (or subroutine or procedure) names all follow the same rules about starting with a letter, etc., and we define each of them to be the same as the syntactical variable < identifier >.

Sometimes, in a definition of this type, there will be more than one alternative. For example, let us make a definition of "integer" not restricted to unsigned integers. If we already know what an unsigned integer is, we may use the following:

> *An integer is an unsigned integer optionally preceded by a plus sign or a minus sign.*

The conventions which we have used thus far do not allow for the words "optionally" or "preceded by," although "followed by" is permitted. Therefore, let us make an equivalent definition, which is slightly longer:

> *An integer is either: (1) an unsigned integer; or (2) a plus sign followed by an unsigned integer; or (3) a minus sign followed by an unsigned integer.*

Now all we need is a symbol for "or." The symbol we use is the vertical line |. Thus our abbreviated definition is

```
<integer>::=<u.i.>| '+'<u.i.>| '-'<u.i.>
```

where we have used "u.i." for "unsigned integer" in order to keep the definition from running off the end of the line. Actually, this precaution is not necessary. Rules in BNF, just like statements in ALGOL, may run to several lines, and position on a given line is immaterial, although, in practice, a definition will be started at the beginning of a new line. Thus

```
< integer > ::= < unsigned integer >
        | '+' < unsigned integer >
        | '–' < unsigned integer >
```

is a self-contained BNF rule equivalent to the one given above.

The vertical line is often used for "lowest-level" definitions, in which a syntactical variable is being defined as any one of a certain collection of characters. Thus

```
< digit > ::= '0' | '1' | '2' | '3'
        | '4' | '5' | '6'
        | '7' | '8' | '9'
```

is a very common definition. Note that this defines only a single digit, not an arbitrary integer; 63, for example, is not a digit by this definition. We may, if we wish, define "letter" in the same way, as any one of the 26 letters of the alphabet. We may even define "alphanumeric character" as any one of 36 different symbols, although what is usually done is to define "letter" and "digit" first, and then to define

```
< alphanumeric character > ::= < letter >
        | < digit >
```

The definition of an integer, or of an identifier, is slightly more complex. An un-

Figure 1: An identifier which has six characters or less.

< an > ::= < alphanumeric character >

< identifier > ::= < letter >

| < letter > < an >

| < letter > < an > < an >

| < letter > < an > < an > < an >

| < letter > < an > < an > < an > < an >

| < letter > < an > < an > < an > < an > < an >

In order to construct language elements of unlimited length, the trick is to use *recursive* definitions: the quantity being defined is part of the definition itself.

signed *two-digit* integer may be defined very simply

< unsigned two-digit integer > ::=

< digit > < digit >

Similarly, an identifier containing exactly two characters may be defined

< 2-character identifier > ::=

<letter> <alphanumeric character>

Extensions to three characters, four characters, etc., are easy enough to visualize; and now, using the vertical line and a few auxiliary abbreviations, we may put together a definition of an identifier which has six characters or less (see figure 1). In a similar way, we may construct a definition of an unsigned integer containing at most 11 digits, or however many digits are permitted on a given computer.

This kind of construction, however, fails when we do not wish to put *any* limit whatsoever on the number of digits in an unsigned integer or on the number of characters in an identifier. In addition, it is overly cumbersome even in the form given above. Therefore, we must call on some new resource. This has actually been done, historically, in two different ways: One way is designed for languages such as ALGOL, LISP, and SNOBOL, in which most constructions do not have length limitations. The other way is intended for FORTRAN and for simplified versions of ALGOL, as well as for various other languages, in which length limitations do exist. We shall consider these in historical order.

The response of the ALGOL group to this problem was to use the resources of mathematics, which rescue us (as they do so often) with what looks like magic. The trick is to use *recursive* definitions, which use the quantity being defined in the definition itself. Consider, for example, the following definition

< unsigned integer > ::=

< digit > < unsigned integer >

| < digit >

Those without a background in mathematical logic may need a considerable amount of time to convince themselves that this definition of "unsigned integer" defines that syntactical variable, in a perfectly valid manner, to be a sequence of digits of any length whatsoever. The argument goes as follows:

1. A digit is an unsigned integer by the above definition.

2. A two-digit number is a digit followed by another digit, and the second digit, by the sentence above, is an unsigned integer. Therefore a two-digit number is an unsigned integer.

3. A three-digit number is a digit followed by a two-digit number; a two-digit number is an unsigned integer by the previous sentence; therefore, a three-digit number is an unsigned integer.

4. A four-digit number is a digit followed by a three-digit number, and so on; the argument may thus be extended indefinitely, with each sentence being used in the proof of the next.

Another common recursive definition is

< identifier > ::= < letter >

| < identifier > < letter >

| < identifier > < digit >

This one is actually easier to understand if the last two alternatives are combined into a single alternative, < identifier > < an >, where < an > means "alphanumeric character" and is defined as either a letter or a digit. Using this syntactical variable, we may rewrite the definition of an identifier as

< identifier > ::= < letter >

| < identifier > < an >

That this constitutes a valid definition may be seen as follows:

1. A letter is an identifier by the above definition.

2. An identifier with two characters consists of a letter, which is an identifier, followed by an alphanumeric character. Therefore, by the second part of the above definition, it is an identifier.

3. An identifier with three characters consists of an identifier with two characters followed by an alphanumeric character. Therefore, by the definition above, it is an identifier. This argument may then be extended to identifiers with four, five, or any number of characters.

Note that this definition of an identifier involves various other identifiers whose names are contained within it. Thus the word TAU is an identifier partly because T is an identifier and so is TA. This fact may lead to confusion because we are constantly reminded, when studying programming languages, that the individual characters in an identifier have no separate meaning. Thus TAU is not (by definition) T times A times U, or TA times U, or T times AU. Nevertheless, TAU is a properly *formed* identifier because T and TA are – just as 2PI is *not* a properly formed identifier because 2 and 2P are not.

Still another common recursive definition is

< argument list > ::=

< expression > ',' < argument list >

| < expression >

This defines an argument list to be a series of expressions separated by commas. It may be used in various ways; for example, the BNF rule

< function reference > ::=

< function name > '(' < argument list > ')'

is one way of defining a function reference (i.e., a use of a function, such as SIN(T*U–B) or ATAN2(X2–X1,Y2–Y1).

The justification for the definition of an argument list is:

1. A single expression is an argument list.
2. Two expressions separated by one comma may be thought of as the first expression followed by a comma followed by an argument list – namely the second expression. Therefore, this is an argument list.
3. Three expressions separated by commas may be thought of as the first expression followed by a comma followed by what remains – namely, the second and third expressions separated by a comma. This much is an argument list, by the sentence above. Therefore, three expressions separated by commas constitute an argument list. The same argument may be used for four, five, etc., expressions separated by commas.

The recursive examples given above are written in what may be called "pure" BNF. The alternative is to add some new conventions to BNF which take care of the recursive cases. This brings us to the second possible response to the problem of representing sequences in BNF. For example, an unsigned integer which is a sequence of from 1 to 13 digits might be written

$$\langle \text{unsigned integer} \rangle ::= \langle \text{digit} \rangle_{1}^{13}$$

and an unsigned integer which is a sequence of an arbitrary number of digits (at least one) might be written

$$\langle \text{unsigned integer} \rangle ::= \langle \text{digit} \rangle_{1}^{\infty}$$

Similarly, an identifier of arbitrary length is given by

$$\langle \text{identifier} \rangle ::= \langle \text{letter} \rangle \langle \text{alphanumeric character} \rangle_{0}^{\infty}$$

Thus a subscript after any syntactical variable stands for a minimum number of repetitions, while a superscript after such a variable stands for a maximum number of repetitions.

The use of subscripts and superscripts in this way solves two problems at once. It makes syntactical variables whose lengths are strictly bounded much easier to represent in BNF. Also, by replacing many of the recursive uses of BNF with nonrecursive uses, it frees the user from having to "think out" these recursive definitions. Nevertheless, the subscript notation does not allow us to eliminate *all* recursion. This will become clear in the examples which we now consider, in which expressions, such as algebraic expressions, are defined in BNF.

There are many types of expressions in programming languages. In ALGOL we have conditional expressions, relational expressions, and Boolean expressions, in addition to the standard simple arithmetic expression. Some of these are easy to define in terms of others. For example, a relational expression (such as $P > Q$ in an *if* statement) is defined by

< relational expression > ::=

< arithmetic expression >

< relational operator >

< arithmetic expression >

where a relational operator may be any one

INTRODUCING
DUAL DRIVE MINIFLOPPY FOR PET!



DEALER INQUIRIES INVITED

DKH641 IS A PRODUCT OF COMPUTHINK.

- *DUAL MINI FLOPPY DRIVE WITH 100K PER DISK SIDE FOR TOTAL 200K ON LINE.
- *DESIGNED FOR COMMERCIAL SPEED REQUIREMENTS *FAST LOADING SPEED.
- *DISKMONTM (DOS) AUTOMATICALLY REORGANIZES FREE DISK SPACE AFTER SAVE OR ERASE.
- *DISKMON IS RESIDENT IN ROM VIA DISK CONTROLLER BOARD PLUGGED INTO EXPANDAPET.*
- *DISKMON ADDS 16 COMMANDS TO BASIC INCLUDING DISK DATA FILES.
- *DISKMON COMMANDS SUPPORT COMMERCIAL PRINTER OFF PARALLEL PORT SUCH AS CENTRONICS 779.
- *FULL DISK SOFTWARE SUPPORT * FORTRAN & PLM COMPILERS THIS JANUARY.
- *90 DAY MANUFACTURER'S WARRANTY ON HARDWARE*READY TO USE ON DELIVERY. WITH FULL INSTRUCTIONS AND UTILITY DISKETTE.
- *CALL OR WRITE FOR ADDITIONAL INFORMATION * INITIAL QUANTITIES LIMITED.
- *THIS SYSTEM IS FULLY SUPPORTED BY SOFTWARE FROM THE MANUFACTURER.
- *ORDERS ARE NOW BEING ACCEPTED FOR DECEMBER & JANUARY DELIVERY.

DKH642	—DUAL DRIVE SYSTEM, COMPLETE WITH DISKMON	$1295.00
DKL067	—DISKMON ASSEMBLER LISTING/DOS	$19.95
ASM789T	—PET ASSEMBLER ON CASSETTE	$19.95
ASM789D	—PET ASSEMBLER ON DISKETTE (5.5 inch)	$49.95
LNK456	—AUTOLINK LINKING LOADER ON DISKETTE	$49.95
FOR300	—FORTRAN COMPILER ON DISKETTE (JAN'79)	$69.95
PLM400	—PLM COMPILER ON DISKETTE (JAN'79)	$49.95

★ ★ ★ BUSINESS PACKAGES STARTING IN 1st QUARTER 1979 ★ ★ ★

*THIS SYSTEM REQUIRES EXPANDAPET MEMORY (MINIMUM 16K—SEE BELOW)

PET COMPUTER

If you haven't bought a pet yet, call for our pet info package!



WHY NOT BUY FROM THE BEST?

8K PET	$ 795
24K PET (8 + 16K)	$1220
32K PET (8 + 24K)	$1320

ALL PRICES INCLUDE 48 HR. PRE-SHIPMENT TESTING & 3 FREE CASSETTE PROGRAMS

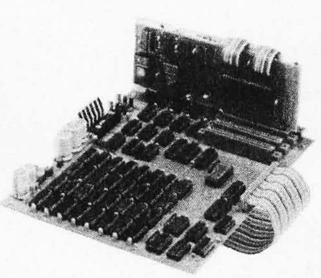
PRICES SHOWN ABOVE INCLUDE EXPANDAPET. PME MEMORIES WILL HAVE HIGHER PRICING.

- ★ NEECO IS A CUSTOMER ORIENTED, FULL SERVICE COMPANY.
- ★ PETS RECEIVE 48 HR. 'BURNIN' BY NEECO BEFORE SHIPMENT.
- ★ FULL CUSTOMER SERVICE AND FULL PRODUCT SUPPORT.
- ★ 48 HR MAXIMUM 'TURNAROUND' ON PET WARRANTY SERVICE ON PETS PURCHASED FROM NEECO.
- ★ FULL PRE-PURCHASE INFO AVAILABLE FROM OUR PET INFO PACKAGE — WE ANSWER CUSTOMER QUESTIONS!
- ★ AUTOMATIC SOFTWARE/HARDWARE UPDATES VIA OUR PET OWNERS MAILING LIST — CALL/WRITE TO BE LISTED!
- ★ COMMERCIAL QUANTITIES AVAILABLE.
- ★ WE ALSO MARKET REPLACEMENT RAMS & ROMS, ETC.
- ★ OFF THE SHELF DELIVERIES (NO DEPOSIT REQUIRED).

SCHOOL INQUIRIES INVITED!

INTERNAL MEMORY EXPANSION FOR PET!

EXPANDAPET™ *INTERNAL MEMORY EXPANSION UNIT*



DEALER INQUIRIES INVITED

- *MOUNTS EASILY INSIDE YOUR PET
- *EASY TO INSTALL (15 MINUTES)
- *NO DEGRADATION OF PET SYSTEM
- *USES LOW POWER DYNAMIC RAMS
- *90 DAY PART&LABOR, 1 YR-RAMS.
- *30 DAY MONEY BACK GUARANTEE.
- *MOUNTING SLOTS FOR 4 BOARDS.
- *CALL/WRITE FOR ADDITIONAL INFO
- *DEALER INQUIRES INVITED.

EXPANDAPET PRICES

16K (+8K PET = 24K)	$425
24K (+8K PET = 32K)	$525
32K (+8K PET = 40K)	$615

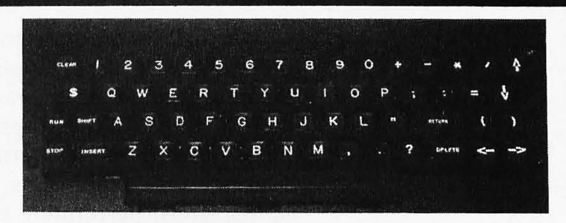
OPTIONAL PLUG-IN BOARDS

SERIAL I/O BOARD	$75
S-100 I/O BOARD	$75
4K EPROM BOARD	$50

32K UNIT ALLOWS 8K OF ASSEMBLY LANGUAGE SUBROUTINES ACCESSED VIA THE USR COMMAND.

EXPANDAPET IS A PRODUCT OF COMPUTHINK.

FULLSIZED TYPEWRITER KEYBOARD FOR PET!



- *COMMERCIAL QUALITY KEYBOARD WITH METAL ENCLOSURE.
- *BASIC TYPEWRITER DESIGN FOR TOUCHTYPISTS.
- *SINGLE KEY FUNCTIONS FOR ALL CURSOR CONTROLS. SHIFT/RUN, INSERT, CLEAR SCREEN/HOME CURSOR, MORE.
- *FUNCTIONS SIMULTANEOUSLY WITH PET'S KEYBOARD.
- *PLUGS DIRECTLY INTO PET'S LOGIC BOARD.
- *DOES NOT USE USER OR IEEE-488 PORTS.
- *NPK-101 IS FULLY TESTED & READY TO USE.
- *ATTACHES DIRECTLY TO FRONT OF PET'S FRAME.
- *CAN BE USED AS A REMOTE TERMINAL (SPECIAL ORDER).
- *30 DAYS TRIAL PERIOD * 90 DAY WARRANTY.
- *CALL OR WRITE FOR FULL SPECS-INITIAL QTY LIMITED.

ORDERS ARE NOW BEING ACCEPTED FOR DEC/JAN CUSTOMER DELIVERIES. **$139.95**

NPK-101 IS A PRODUCT OF NEW ENGLAND ELECTRONICS.

WE CANNOT LIST ALL OF OUR SOFTWARE AND HARDWARE PRODUCTS
CALL OR WRITE FOR OUR *FREE* SOFTWARE/HARDWARE DIRECTORY

DOMESTIC & OVERSEAS DEALER INQUIRIES INVITED ON * MEMORY * KEYBOARD * FLOPPY

NEECO

NEW ENGLAND ELECTRONICS CO., INC.
248 BRIDGE ST., SPRINGFIELD, MASS. 01103
MON-FRI 9:30-5:30, SAT 10-2, E.S.T.

(413) 739-9626





ACCEPTED, ADD 3% SERVICE CHARGE.

of the six relations (greater than, less than, etc.) expressed in a manner which depends on the language used. In any sort of expression containing operators and possibly nested parentheses, however, the definition will be much more complex. We shall indicate here how to define simple arithmetic expressions in BNF; the basic technique used here may be used in other kinds of expressions.

What is a simple arithmetic expression? Clearly it is *not* simply any combination of identifiers, constants, operators, and parentheses. We may specify certain rules (the parentheses have to balance, an operator cannot be the last character in the expression, etc.) but it is difficult to know when we have specified all of them. One key to defining expressions is obtained by means of the *precedence rules*. You have probably seen these, although possibly not by this name; these are the rules that specify that multiplication is performed before addition, and the like. Thus in order to evaluate an expression (without parentheses) there are three basic steps:

One key to defining the BNF notation of expressions is an understanding of operator precedence rules. These rules can often be made a part of the BNF grammar of a language.

1. Perform all the exponentiations.
2. Perform all the multiplications and divisions.
3. Perform all the additions and subtractions.

We shall incorporate these steps into our definition of an expression by defining four separate syntactical variables: primary, factor, term, and expression. Factors are made up of primaries; terms are made up of factors; and expressions are made up of terms. Thus, performing all the exponentiations in an expression corresponds to grouping the primaries into factors, and similarly for the other two steps mentioned above.

In order to define a factor as a collection of primaries separated by exponentiation signs, we note that this is similar to defining an argument list as a set of expressions separated by commas. We need only substitute "exponentiation sign" for "comma," and "primary" for "expression." Thus the definition is

< factor > ::= < primary > '↑' < factor >
    | < primary >

We could also rewrite this definition in the other mode

< factor > ::= < primary > ['↑' < primary >]$_0^\infty$

In other words, a factor is a primary followed by any number of occurrences, including none, of an up-arrow (↑) followed by another primary. This illustrates another feature of extended BNF: the use of *square brackets*, the signs [and]. Square brackets in BNF serve roughly the same function as parentheses do in algebra, and the use of square brackets in BNF is sometimes called *factoring*.

We continue our definition of an expression by defining a term as a collection of factors separated by multiplication and division signs. One way to do this is to define a "multiplication operator," or "mulop," as *either* * or /. The definition can then take the same form as before (we illustrate here only the recursive formulation)

< term > ::= < factor > < mulop > < term >
    | < factor >

An expression would then be defined in a similar way, using "addop" for either + or —

< expression > ::=
    < term > < addop > < expression >
    | < term >

where we have used "expression" as short for "simple arithmetic expression." Alternatively, both of these could be written out

< term > ::= < factor > '*' < term >
    | < factor > '/' < term >
    | < factor >

< expression > ::= < term > '+' < expression >
    | < term > '—' < expression >
    | < term >

The only question remaining is what we mean by "primary." This differs from one language to another; roughly speaking, a primary is one of the "elementary" constructions which are connected by the operators, such as a constant, a variable, a subscripted variable, or a function reference. There is always, however, one special type of "primary" which takes care of parentheses.

Up to now we have considered only expressions without parentheses. In one sense, when we introduce parentheses into an expression, we sometimes violate the precedence rules upon which we have built

the entire preceding construction. Thus in the expression

```
A-B*C/(D-E)
```

we do *not* perform all the multiplications and divisions first. In fact, the subtraction D–E must be performed before the division of C by the result.

This expression, however, may also be thought of as

```
A-B*C/F
```

where F stands for (D–E). In this new expression, we do perform all the multiplications and divisions before the additions and subtractions. The process of substituting F for (D–E) may suggest to us that *an expression in parentheses can be treated as if it were a single primary*. This is also true for more than one level of parentheses. Thus, the expression

```
A+B*(C-D*(E+F/G)-H)+I
```

may be thought of as the expression

```
A+B*J+I
```

where J stands for the expression

```
C-D*K-H
```

in which K stands for the expression E+F/G. In each of these three expressions – E+F/G, C-D*K-H, and A+B*J+I – the precedence rules apply; and the last two of these contain primaries (K and J) that take the form of an entire expression in parentheses. The BNF description of an expression is now completed by adding this form of a primary to the definition. Thus

```
<primary> ::= <constant> |variable>
    | <array reference> | <function reference>
    | '(' <expression> ')'
```

is a sample definition of a primary in which the syntactical variables "constant," "variable," etc., should be further defined according to the rules of the particular language under consideration.

One final and important fact about BNF is that although widely used it is *not* universal enough to describe the syntax of every well-known programming language. In particular, COBOL (Common Business Oriented Language) contains certain constructions which are not amenable to BNF. The formal definition of the PL-I language (see reference 5) is made according to a separate set of abbreviation conventions which are similar, but not identical, to BNF.■

REFERENCES

1. Naur, P, ed, "Report on the Algorithmic Language ALGO 60," *Communications of the ACM*, 3 (May 1960), pp. 299-314.
2. Knuth, D E, *Semantics of Context-Free Languages*, Math Systems Theory, 2, 2 (1968), pp. 127-145.
3. Knuth, D E, *On the Translation of Languages from Left to Right*, Inf Contr 8 (Oct 1965), pp. 607-639.
4. Wirth, N, and Weber, H, "Euler, A Generalization of ALGOL, and Its Formal Definition," *Communications of the ACM*, 9 (January and February 1966), pp. 13-25 and 89-99.
5. IBM Form C28-6571-4, *IBM System/360 Operating System, PL-I: Language Specifications.*

Creating a Chess Player

Peter W Frey
Dept of Psychology
Northwestern University
Evanston IL 60201

Larry R Atkin
Health Information Services
542 Michigan Av
Evanston IL 60202

The chess program that we have presented in parts 2 and 3 of this series (November 1978 and December 1978 BYTE, pages 162 and 140, respectively) represents a modern implementation of the basic type A strategy described by Shannon in 1950 (see references). If run on a powerful computer, this type of program can play a reasonably good game of chess. Its major weakness lies in its inability to engage in long-range planning. In many middle and end game positions, it will make seemingly aimless moves. Once it attains a position which optimizes the general heuristic goals of its evaluation function, it is faced with the prospect of finding a move which alters the position as little as possible. If the opponent is skillful in developing a long-range attack while not providing any immediate targets, the machine may simply shuffle its pieces back and forth until its position becomes hopeless. The absence of reasonable goal directed behavior is a common limitation of problem solving techniques which are based solely on forward search. The solution of this problem would have important implications for a wide variety of artificial intelligence tasks.

To play a strong game of chess, it is necessary to have a plan. To have a plan, however, the program must recognize specific patterns and relate them to appropriate goals. This, in turn, requires that the program have access to the detailed kind of chess knowledge which is characteristic of the skilled human player. Thus, we seem to have come round in a circle. In order to avoid selective searching, we have adopted a strategy which does not require very much chess knowledge. In examining the weaknesses of this approach, we discover that the forward search can only be truly successful if we have a clear idea of what we are looking for. To know what we are looking for, however, we must have more knowledge about chess.

So where do we go from here? The highly skilled players who are familiar with the chess programming literature (notably, Berliner, Botvinnik and Levy) are unanimous in their enthusiasm for a selective search strategy. Berliner (see references), for example, advocates a procedure in which very small (for a computer) look-ahead trees are generated, eg: 200 to 500 nodes. His idea is that the program should make an intensive analysis at each node "in order to ascertain the truth about each issue dealt with." Chess knowledge should play a primary role in directing the tree search. The search itself would discover additional relevant information and this would provide an even more knowledgeable focus for the search. This procedure is analogous to the progressive deepening technique which de Groot discovered in the human grandmaster and is the exact antithesis of the brute force (type A) strategy (see October 1978 BYTE, "Creating a Chess Player, An Essay on Human and Computer Chess Skill," page 182).

The efforts of the last decade have demonstrated that the selective search strategy is harder to implement than the full-width approach. In addition, full-width searching has consistently produced superior chess. Despite this, there is hardly anyone familiar with chess programming who does not believe that further progress depends on increasing the amount of chess knowledge in the program. The key question is not whether this should be done but how to do it. Since the selective search approach has not led to notable progress, perhaps it is time to consider a different approach.

We believe that a viable alternative exists which combines the proven virtues of the full-width procedure with the potential advantage of a goal-directed search. The central idea is the development of a unique evaluation function for each position. In addition to the general heuristics which are presently

employed, evaluations should consider features which are germane to appropriate goals.

According to this plan, move selection would involve two separate stages. In the first phase, a static analysis of the position would be made in an attempt to discover key patterns. This process would involve a hierarchical analysis in which the features of the position would be compared with a general set of library patterns. Highly specific features would be identified and relevant chess-specific knowledge would be accessed. This information, including appropriate short term and long term goals, would be used to construct a conditional evaluation function which would assess the usual general features (eg: material, mobility, King safety, etc) and also other features which are meaningful only in specific situations. Once the conditional evaluation function has been constructed, the second phase of analysis would begin, a conventional full-width tree search employing the special evaluation function.

The first phase of this process would rely heavily on domain specific knowledge (ie: information about chess). It would require a pattern recognition facility and an organizational plan for storing a vast amount of chess knowledge in a manner conducive to rapid retrieval. When this first phase was successful in identifying appropriate goals and producing relevant modifications in the evaluation function, the full-width search which followed would select a move which was thematic with the appropriate goal. If the first phase were unable to identify a key feature, the evaluation function would employ the same general heuristics which it presently uses. For this reason, the pattern recognition and information retrieval modules can be gradually implemented without a lengthy period in which serious blunders are frequent occurrences. This is a major advantage that the conditional evaluation function has in comparison to a selective search strategy.

Chess Structure

To implement a conditional evaluation function, it is necessary to develop a hierarchical descriptive structure for chess. At the top level, one can make the conventional distinctions between the opening, the middle game, and the end game. Within each of these three major divisions, there would be many specific subdivisions. Within each subdivision, there would be many specific variations.

The opening has three major themes: to develop a pawn structure which is favorable for you but unfavorable for your opponent; to increase the mobility of your minor pieces and limit the mobility of your opponent's minor pieces; and to castle as soon as possible and delay your opponent's opportunity to castle. These general goals provide a framework for evaluating specific variations. They do not provide a specific prescription for selecting a move because a sequence of moves which is thematic with these goals may have a tactical refutation. An apparently good move may not work because it loses material. For this reason, general principles are best applied at the terminal points of a look-ahead search rather than being used as a checklist for selecting the most thematic move as advocated by Church and Church in *Chess Skill in Man and Machine* (see references).

Part 4

Strategy in Computer Chess

The tournament player who knows opening theory as well as many specific move variations will have a clear advantage over an opponent who knows the general principles but is not familiar with the specific variations. For this reason, tournament players and good chess programs rely on a library of memorized opening variations. The contestant who has carefully planned his opening variations can often gain an important advantage early in the game. To maximize the benefit of a well-prepared opening library, it is also necessary to continue the general theme of the opening once the predigested move sequences have been exhausted. At this stage it is necessary to have a conditional evaluation function. When the machine leaves the library and starts to use a look-ahead procedure to calculate its move, it should use an evaluation function that augments general opening principles with special goals which are thematic with that type of opening.

A portion of the work required to implement this proposal has already been started. Chess specialists have prepared highly detailed analyses of specific opening variations and have developed well-defined rules for categorizing different move

sequences into specific subdivisions. For example, a game which starts (1) P-K4, P-K3 is labeled as the French defense. If the game continues (2) P-Q4, P-Q4; (3) N-QB3, B-N5, it is called the Nimzovich (or Winawer) variation of the French defense. If it continues (2) P-Q4, P-Q4; (3) N-QB3, N-KB3, the game is labeled as the classical variation. A continuation of (2) P-Q4, P-Q4; (3) N-QB3, PxP is called either the Rubinstein variation or the Burn variation depending upon subsequent moves. A different approach develops from (2) P-Q4, P-Q4; (3) N-Q2, which is labeled as the Tarrasch variation. And there are many more. The important point, however, is that each of these variations can be objectively identified, and that for each there are well-developed strategical ideas and specific immediate goals. These ideas can be stored in the opening library and can be retrieved when the machine leaves the library. In addition to general opening heuristics, the evaluation function would reflect the specific theoretical ideas which are appropriate to the particular opening at hand. In principle, this idea can be implemented without difficulty. In practice, however, a tremendous amount of chess knowledge is needed and hours and hours of effort are required. To our knowledge no serious attempt has yet been made to implement this strategy. The information on opening theory is needed only once during a game and thus could be stored on disk, since rapid access is not critical.

Pattern Recognition and the Middle Game

From a conceptual point of view, the application of chess knowledge to the evaluation function in the middle game is much more challenging. In this case, pattern recognition becomes an important ingredient. In implementing a goal oriented move selection strategy, Church and Church limited their middle game strategy to either a Kingside attack, a Queenside attack, or concentration on a weak point (ie: a target). The Kingside or Queenside attack is triggered when the machine determines that it has superior forces on one side or the other. This determination can be based on who controls key squares. In calculating the power relationship of different pieces over given squares, it is important to note that less valuable pieces exert more control than valuable pieces. A pawn has greater control over territory than a Queen because it is harder to dislodge. If an attack on one side or the other is deemed appropriate, the evaluation function can be modified to give an extra bonus for moves which augment the attack on that side and for moves which increase the pressure on critical squares.

Pattern analysis is also important in detecting an appropriate target. There are several well-known chess relations which provide obvious targets for attack. One is the *backward pawn* which is prevented from advancing by a pawn or a minor piece. Another natural target is the minor piece which is pinned to the King or Queen. The third is the overworked piece, a key element in the defense against two or more different attacks. If the latter is removed in an exchange, the pieces it is defending will be open for attack. A fourth natural target is a square which would permit a Knight to fork two major pieces (ie: Rook, Queen, King) or a Bishop to skewer two major pieces. If the machine threatens to control that square and to locate an appropriate piece there, the opponent will be forced to devise a defense. Once one of these targets has been detected, the evaluation function can be modified to give a bonus for moves directed at the target. In addition, a plan might be devised to encourage the use of a decoy (a pawn or minor piece which is sacrificed to bring an important piece to a particular square) or to capture a piece which is serving an important defensive function.

A Chess "Snapshot"

In the past, programmers have attempted to implement such plans by using a selective search (eg: Berliner, Zobrist and Carlson) or by using no search at all (eg: Church and Church). Zobrist and Carlson (see references) have developed an innovative technique in which "computer snapshots" are devised which summarize important piece relationships such as attacks, pins, skewers, forks, etc, which presently exist in the given position, or which could occur after one or two moves. Each snapshot is given a weight based on the relative values of the pieces involved and the location of the pieces in respect to the opposing King and the center of the board. The weighted snapshots are then used to select moves for inclusion in a Shannon type B tree search. This procedure provides considerable goal direction to the move selection process.

Although the Zobrist-Carlson snapshot procedure has much to offer (including a highly efficient bit map implementation strategy), it incorporates a common problem shared by all selective search techniques. Occasionally an important continuation is overlooked and this results in the selection of an inappropriate move which may be a gross blunder. By implementing the plans

derived from the computer snapshots in the form of a conditional evaluation function, instead, the program can benefit from goal directedness without risking the oversights which are characteristic of selective searching. In this way, the machine can retain the benefits of the full-width search and at the same time engage in strategic planning.

There is a special class of positions for which this approach is especially appropriate. In his thesis at Carnegie-Mellon University Berliner described a special problem, the horizon effect, which plagues the conventional look-ahead approach (see *Chess Skill in Man and Machine,* pages 73 thru 77). One version of this problem involves a piece which is trapped and cannot be saved. Forward searching programs often engage in a bit of foolishness by making forcing but poor moves (such as attacking pieces with pawns or sacrificing pawns for no advantage) which delay the capture of the trapped piece and push its eventual loss beyond the horizon of the tree search. By doing this, the program erroneously concludes that the piece is safe, when in reality the planned move sequence weakens a reasonable position and is still insufficient to save the piece. In this type of situation, the trapped piece should be given up for lost and the program should do its best to take advantage of the tempo required by the opponent to capture the piece. A piece whose time has come is sometimes referred to as a desperado. The only option available is to make the opponent pay as dearly as possible for the capture. If the desperado can be traded for a pawn or a piece of lesser value, this is preferable to being given up for nothing.

This strategy can be implemented with a conditional evaluation function by simply assuming that the trapped piece has a material value of zero. This change would cause the search process to trade the piece for the highest valued candidate that can be found. This is obviously better than having the program engage in useless sacrifices of position and material in a hopeless attempt to resurrect a lost piece. The key element to this implementation is the ability to determine when a piece is truly lost and can be labeled as a desperado. This is a very difficult problem even for a very sophisticated pattern analysis facility.

End Game Considerations

The most interesting application of the conditional evaluation function is in the end game. Because end game strategy is highly dependent on the specific characteristics of the position, a general purpose evaluation function is not very effective. It is necessary to understand what is required in a given position and then select moves which are clearly directed at an appropriate goal. Church and Church list three common goals in the end game: to mate the opponent's King, capture a weak pawn, or promote a pawn. In this case, pattern analysis is important. First the machine must be able to identify the position as one belonging to the end game. Then it has to determine whether a mate attempt is reasonable or whether a pawn can be captured or promoted. Church and Church (see *Chess Skill in Man and Machine*, pages 151 thru 154) describe a general strategy for identifying and capturing a weak pawn. Although their approach does not involve a forward tree search, the specific techniques which they describe can be adapted to the full-width search strategy. Let us consider several specific end game positions involving either a mate, a pawn capture, or a pawn promotion.

For a number of mating situations, a specific algorithm (step-by-step instructions) or a complete lookup table can be developed to produce mate in a minimum number of moves. Typical applications would be King and Queen versus King; King and Rook versus King; and King, Bishop, and Knight versus King. The mating algorithm for each case would include rules for assigning the potential piece relationships into a few general categories, and a prescription for an appropriate type of move for each category. This approach requires no search. A second approach involving a lookup table is even more explicit. An appropriate move is stored in a table for every possible piece configuration. To play the mate perfectly, the machine uses the position to determine an address in the table and then simply reads the correct move.

Both of these procedures are perfectly feasible and avoid many problems which can be encountered in the end game. The limitation of this approach is that there are a very large number of mating situations and a tremendous amount of work would be required to make a detailed analysis of each one. In addition, this strategy requires the storage of a great deal of information which would be used only infrequently.

A third approach, and one which is thematic with the idea of conditional evaluations, is to make a small modification in the evaluation function for each specific mating situation. The notion is that a shallow search combined with a few key ideas should suffice to produce a mate in a reasonable number of moves. With King and Queen or King and Rook versus King, it is sufficient for the

program to "know" that the defending King must be forced to the edge. To do this, the program simply needs to add bonus points to the evaluation function when the defending King is near the edge. The size of the bonus should be a linear function of closeness to the edge. This modification of the evaluation function causes the minimax search to select a pathway in the look-ahead tree which forces the defending King to the edge.

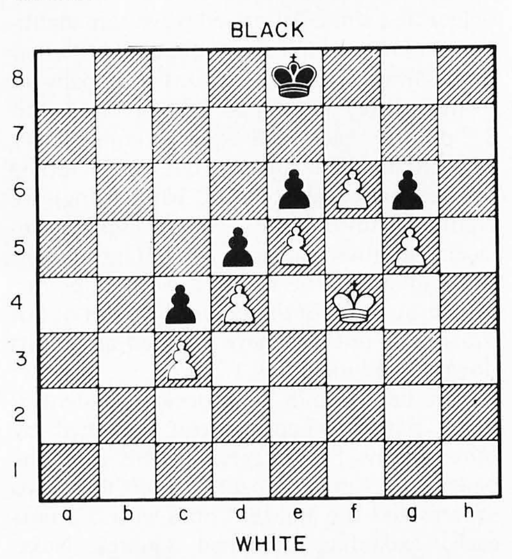
With King, Bishop, and Knight against King, the job is slightly more complicated. In this case it is important to know that the defending King must be forced to one of the two corners having the same color as the Bishop's squares. The trick is to add a large bonus when the defending King is on the appropriate corner squares and a smaller credit when it is near these corners. This modification will cause the minimax procedure to find a sequence of moves which forces the defending King into one of the appropriate corners. The general theme is that the full-width search is a powerful device by itself and that the addition of a small amount of chess knowledge is sufficient to produce the desired outcome.

Kings and Pawns in the End Game

Some of the most challenging positions in the end game involve only Kings and pawns. Many of these require an approach which is more sophisticated than those described previously. Consider, for example, the position diagrammed in figure 1. This is a modification of a position presented in Berliner's thesis which demonstrates one of the major weaknesses of a full-width forward search. White has a pawn on f6 which could advance and be promoted if the Black King were out of the way. *[Algebraic notation is used throughout this article to designate chessboard squares. The horizontal rows (ranks) are numbered from 1 to 8, starting at the bottom (White). The files are labeled a through h from left to right . . . .* CM*]* To win, White must do an end run with his King and bring it to the aid of the pawn. Since Black cannot attack White's pawns on c3 or g5 without leaving the passed pawn, he is helpless to stop White's maneuver. Although this analysis is obvious at a glance to an experienced player, a program that discovers truth by doing a full-width search is faced with a difficult problem. In order to determine that the King can force promotion of the pawn, White must complete a look-ahead search of approximately 35 plies. This is beyond the scope of even the most powerful computer. If the machine employs a general purpose algorithm which encourages the King to centralize its position during the end game, it will search for a pathway which eventually places it on its present square (f4) or one of the neighboring squares (e3 or f3). Because of this, the correct sequence of moves would never be discovered.

In order for a full-width search to make progress in this type of position, the evaluation function must produce goal direction. One way to do this is to provide a bonus for moves which reduce the distance between the White King and the passed pawn. A secondary goal is to reduce the distance between the White King and any Black pawns which are not defended by another pawn. A tertiary goal is to centralize the White King. The first step in developing a specific implementation of this plan is to identify the territory which is denied to the White King. For this purpose, we wish to determine which squares are controlled by the pawns. The White King cannot move to a square occupied by one of its own pawns, nor can it move to a square attacked by an opposing pawn. Figure 2 presents a map of the position with each of the forbidden squares darkened. The location of these

Figure 1: Chess position which demonstrates a weakness of the full-width forward search. *In this example, White has a pawn on square f6 which could advance and be promoted if the Black King were out of the way. To win, the White King must come to the aid of the pawn. Since Black cannot attack White's pawns on c3 or g5 without leaving the passed pawn, he is helpless to stop White's maneuver. Although this analysis is obvious to an experienced player, a program using a full-width search would have to search its decision tree to a depth of 35 plies (ie: 35 half moves; a ply is defined as a move by one side) in order to come to the same conclusion.*



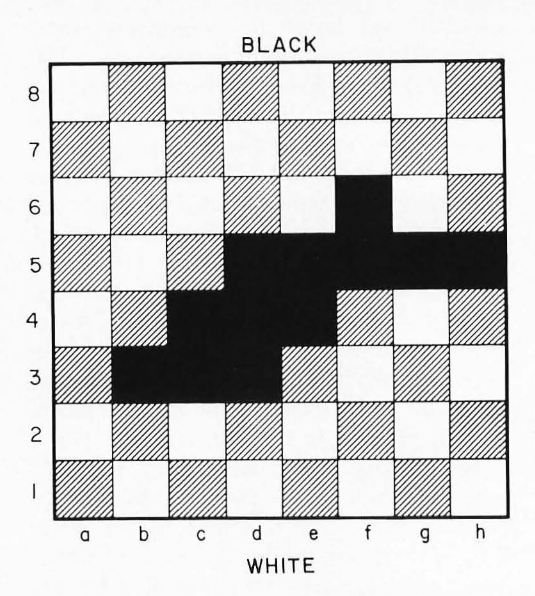
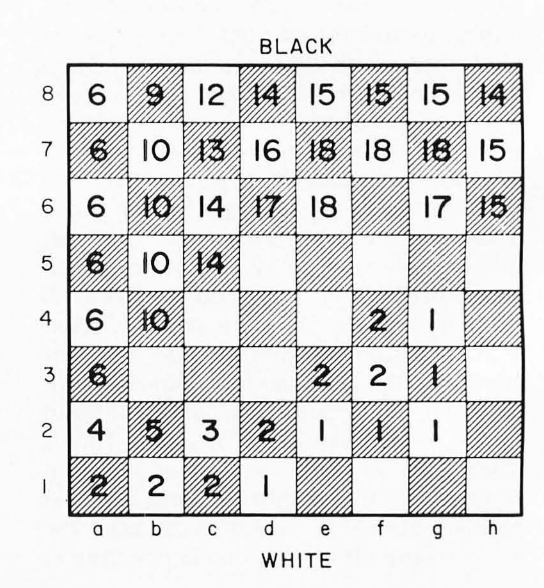


Figure 2: Forbidden squares in the figure 1 position used to help White (the computer) evaluate the position more efficiently. The White King cannot move to a square occupied by one of its own pawns, nor can it move to a square attacked by an opposing pawn. All of these squares are darkened in the figure. This diagram is used in implementing the goal directed technique described by Church and Church (see figure 3).

"taboo" squares provides the defining boundaries for potential access routes to the desired goals. The second step in implementing this plan is to use a technique described by Church and Church. Starting at each goal object, work backward toward the attacking piece(s). In our case, we are interested in creating a reward gradient which encourages the White King to approach its own passed pawn and the target pawns. To do this, we consider one goal object at a time. All passed pawns are identified. In our example, only the White pawn at f6 qualifies. The two squares diagonally in front of it (e7 and g7) are each credited with 8 "points" each. All squares immediately adjacent to these squares (but not including squares inaccessible to the White King) are credited with 7 points. Next all squares adjacent to these squares (excluding inaccessible squares) are credited with 6 points. This process is continued until we run out of squares or until we have assigned all credits down to and including 1.

The next step in the process is to identify Black pawns which are not defended by other pawns (ie: targets). In this case, the pawns at e6 and g6 qualify. Credit these two squares and the adjacent ones with 5 points each, excluding darkened squares. Next, credit squares adjacent to these with 4 points. Continue this process until all available squares have been exhausted or until the value of 1 has been assigned. This process is executed independently for each target pawn. The last step involves credit for centralization. The four most central squares (d4, d5, e4, e5) are credited with 3 points. The squares which surround these squares are credited with 2 points. The squares which surround those squares are credited with 1 point. Points are then removed from any square which is inaccessible to the White King. When this process has been completed, the credits are totaled for each square to provide a bonus map for the White King. This map is presented in

Figure 3: Bonus map for the White King in the position of figure 1, based on a technique described by Church and Church (see references). A goal is established for a particular attacking piece, in this case the White King, and an iterative numerical technique is used to implement it. The goal is to encourage the White King to approach its own passed pawn and the target pawns. (A target pawn is an enemy pawn not defended by other pawns.) Numerical figures of merit are assigned to strategic squares close to White's passed pawn and Black's undefended pawns. Points are also awarded or subtracted for positional characteristics such as centralization of squares, etc. A type of flow algorithm assigns lower and lower values to squares in direct proportion to their distances from the strategic squares, avoiding any forbidden squares. The resulting map of numbered squares enables the King to find the right pathway by constantly searching for ascending values of squares whenever possible.



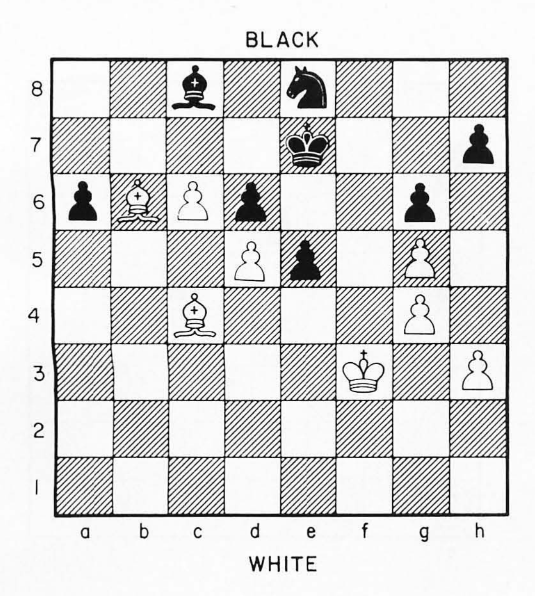


Figure 4: Another end game position, analyzed by the method of Church and Church in figures 5 and 6.

figure 3. By applying this bonus map to the terminal positions of the look-ahead search, the evaluation process will select a move sequence which causes the White King to gravitate in the proper direction. In fact, the correct sequence of moves will be selected even if White is restricted to a 5 ply search each time a move is selected. The bonus map, though simple in concept, has a tremendously beneficial effect.

There is an additional point which needs consideration. In our exposition, we have assumed that the pawns remained stationary. If a pawn were to move, the bonus map would have to be changed. This is not a major problem, however, since there are only a small number of positions that can result from pawn moves, and once the bonus map has been computed for a given configuration, it can be stored and used each time that configuration is encountered in the look-ahead tree. For this reason, the calculations which are required will not be particularly time consuming.

Another example of this strategy is based on the position presented in figure 4. This is a slight modification of figure 6.7 from the chapter of *Chess Skill in Man and Machine* by Church and Church. To apply our technique with respect to the bonus map for the White King it is necessary to determine which squares are not accessible to the White King by virtue of pawn control. As before, these include squares occupied by White pawns and squares attacked by Black pawns. The relevant squares are darkened in figure 5.

The next step is to locate passed pawns for White. There is only one and it is located at c6. The two squares diagonally in front of this pawn (b7 and d7) are credited with 8 points. Squares adjacent to these squares which are not among the darkened squares in figure 5 are credited with 7 points. Squares adjacent to these receive 6 points. This process is continued until there are no more available squares or until the credit value of 1 has been assigned. The next step is to determine whether any Black pawns are potential targets. As before, a target pawn is defined as one which is not defended by a friendly pawn. In the present example, there are three candidates: the pawns at a6, d6 and h7. For each pawn, the value of 5 is credited to the pawn's square and the adjacent squares. Then the value of 4 is credited to each adjacent square. This process of establishing a gradient of decreasing values from 5 down to 1 as distance increases from the target is continued until the last values have been assigned. This is done for each target pawn and in each case, squares darkened in figure 5 are always excluded from the process. The last assignment process is conducted for centralization, with center squares (d4, d5, e4 and e5) receiving 3 credits each and neighboring squares receiving 2 credits. The squares one move in from the edge are assigned the value of 1 and then credits are removed from any square which has been darkened. The final step in developing a bonus map for the White King is to total the credits for each square.

The composite map is presented in figure 6. This set of bonus points will encourage the White King to move in the appropriate direction. Without this strategy

Figure 5: Forbidden squares for the position in figure 4.

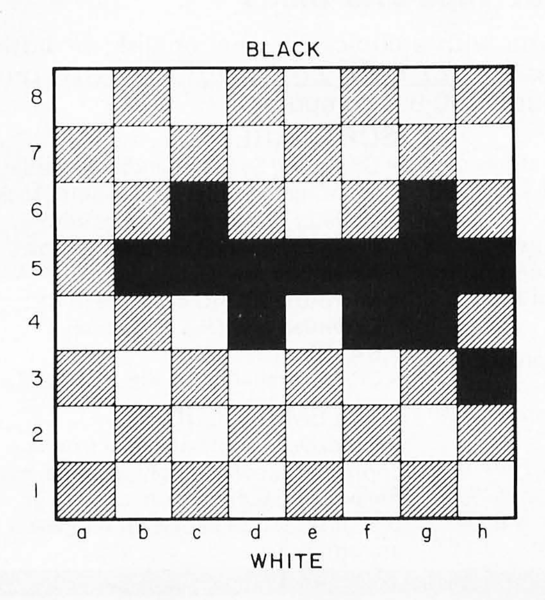


Figure 7: A chess position which can be analyzed efficiently by means of the coordinate square concept proposed by Ken Church (see references). In this approach, the Black King must coordinate precisely with the White King in order to successfully defend its pawns. The technique is illustrated in table 1.

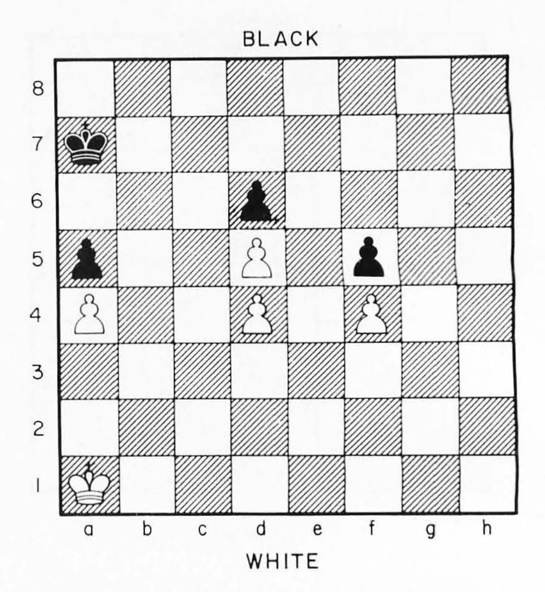


Figure 6: Bonus map for the position of figure 4. Without this map, an 11 ply search would be required for the computer (White) to discover that the pawn at a6 can be captured. Using the map, only a 3 ply search is required.

BLACK

	a	b	c	d	e	f	g	h
8	14	15	16	16	16	15	13	11
7	15	18	18	19	18	16	14	11
6	15	17		19	19	17		11
5	14							
4	11	12	10		6			
3	8	9	10	7	5	3	1	
2	5	6	6	6	4	2	1	
1	3	3	3	3	3	1		

WHITE

an 11 ply search would be required for White to discover that the pawn at a6 can be captured. With the implementation of these attack gradients for the White King, however, the correct move can be selected with only a 3 ply search. As was the case in the previous example, the establishment of a plan within the evaluation function produces

a goal directed search without requiring an enormous look-ahead tree. This increase in efficiency is highly desirable.

Because the process is directed by the location of the pawns, changes in the map will occur infrequently and therefore only a relatively small number of bonus maps will be required for any one search. Once a map has been calculated for a particular pawn configuration, it can be stored and used later whenever it is needed. Although this strategy seems to work well in the examples we have presented, it is reasonable to ask whether this procedure will work in all end game situations. Unfortunately, the answer is no.

Consider the position presented in figure 7. This is a famous end game problem which appears as diagram 70 in Reuben Fine's classic chess book, *Basic Chess Endings* (see references). It was analyzed in 1975 by Monroe Newborn to determine if his special end game program, Peasant, could solve it. After several unsuccessful efforts, Newborn concluded that the problem would require about 25,000 hours of processor time before a solution could be found (see *Chess Skill in Man and Machine*, page 129). The problem is difficult, but not as impossible as Newborn suggests. Because Peasant does not have a transposition table, the program did not take advantage of the tremendous number of identical terminal positions which are encountered when an exhaustive search is made of this position. Because the pawns are locked, the only moves which are possible are King moves, and this greatly increases the potential number of transpositions.

The position was submitted to Northwestern's chess program Chess 4.5 running on the CYBER 176 system at Control Data headquarters in Minneapolis. David Cahlander discovered that Chess 4.5 could solve the problem after a 26 ply search! This required ten minutes of processor time on the powerful CYBER 176. Although it is interesting to know that the problem can be solved by a brute force search, this type of solution is not particularly elegant and it requires a level of hardware sophistication that is not likely to be available in the small system for a few years yet.

The Coordinate Squares Approach

What can be done to make this problem more manageable? Interestingly enough, there is a rather neat approach to problems of this type which has been examined in some detail by Ken Church in his under-

graduate thesis at MIT. Working with Richard Greenblatt as his advisor, Church applied the chess concept of coordinate squares to this position. The basic notion is that the Black King must coordinate precisely with the moves of the White King in order to successfully defend its pawns. For any particular square which the White King occupies, there are only a limited number of squares which the Black King can occupy and still hold his act together.

In his thesis, Ken Church presents a fairly extensive analysis of King and pawn end games. For our present purpose, we will limit our analysis to King and pawn end games in which the pawns are locked and we will modify Church's approach to suit our conditional evaluation strategy. The major difference is that Church attempts to discover a complete solution to the problem using the coordinate squares idea. We propose, instead, to use the coordinate squares approach to provide the evaluation function with additional chess knowledge. With this modification, a full-width search of reasonable depth can find the correct move.

Using figure 7 as an example, the first step in this process is to determine which squares are denied to each of the Kings by the existing pawn configuration. By noting that each King cannot move to a square that is occupied by its own pawn or that is attacked by an opponent's pawn, one can easily determine that squares a4, b4, c5, d4, d5, e4, e5, f4 and g4 are denied to the White King. Likewise, squares a5, b5, c5, c6, d6, e5, e6, f6 and g6 are denied to the Black King. Neither side has a passed pawn, but there are multiple targets, since none of the pawns are defended by friendly pawns.

By applying the strategy described earlier, it is possible to calculate a composite attack map for the White King on the basis of the target pawns at a5, d6, and f5 and taking into account the centralization subgoal. The resulting map for Fine's position is presented in figure 8. The squares without a number are the squares which are denied to the White King because of the pawn structure. Given the position of the White King (a1), a shallow search using this attack map as part of the evaluation function would encourage the White King to approach the target pawn at a5 (eg: b2, c3, c4, b5, a5). If the Black King were more than five moves from a5, this sequence of moves would lead to success. Given that the Black King is at a7, however, this plan is doomed to failure. In fact, the first move in the sequence, b2, is fatal and transforms a winning position into a draw. There are two important conclusions that follow from this discovery. The first is

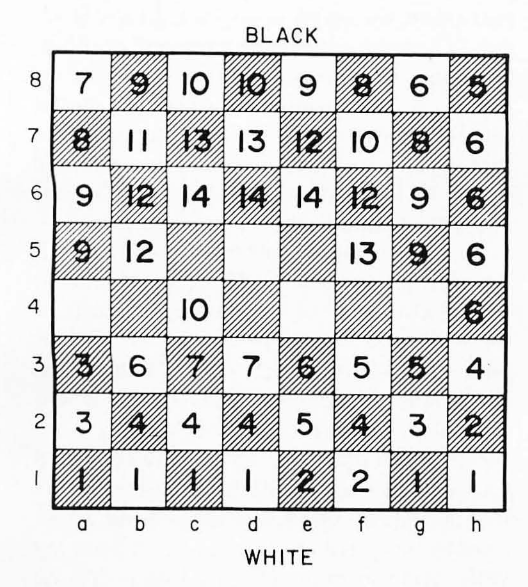


Figure 8: Bonus map for the position of figure 7, a composite attack map for White based on the target pawns at a5, d6 and f5, and taking into account the centralization subgoal.

that our simple goal-gradient approach does not always work. The second is that chess end games are much more difficult than a novice player might suppose.

Let us extend Ken Church's ideas and apply the concept of coordinate squares to this position. First, we wish to assign each of the squares to one of three categories: under the influence of the Black King, under the influence of the White King, or contested. To do this we compute the distance from each King to each square, given the constraints imposed by the existing pawn structure. This creates two distance maps, one for the White King and one for the Black King. For squares which are not accessible to one or both of the Kings, we assign a distance score based on the number of King moves required to reach that square by traveling across accessible squares. Next, each square which is closer in moves to the Black King than to the White King and is not denied to the Black King is assigned to Black. Each square which is closer to the White King than to the Black King and is not denied to the White King is assigned to White. The remaining squares are assigned to the contested category. The results of this procedure are summarized in figure 9. The squares assigned to Black are indicated by the letter B and the squares assigned to White are indicated by a W. The blank squares belong in the contested category.

If the territory under the influence of either King is adjacent to an opponent's pawn, the contest is essentially settled since

Figure 9: The square control concept applied to the position of figure 7. Each of the squares is assigned to one of three categories: under the influence of the Black King, under the influence of the White King, or contested. To do this, the distance from each King to each square is computed, given the constraints imposed by the existing pawn structure. Each square closer in moves to the Black King and not denied to the Black King is assigned to Black, and vice-versa. The remaining squares are labelled as contested. Through a complex series of manipulations and the use of so-called frontier squares (see text), White is actively directed to attack Black's pawns using the strategy of trying to prevent Black from moving onto strategic coordinate squares which are vital to Black's defense.

BLACK

	a	b	c	d	e	f	g	h
8	B	B	B	B	B	B	B	B
7	B	B	B	B	B	B	B	B
6	B	B				B	B	B
5								B
4			W					W
3	W	W	W	W	W	W	W	W
2	W	W	W	W	W	W	W	W
1	W	W	W	W	W	W	W	W

WHITE

that pawn would be open for capture. Since this is not the case for the present position, we wish to define a special category of squares called *frontier squares*. A frontier square is any square under your influence that is adjacent to an accessible contested square or is adjacent to an accessible square under the influence of the opponent.

For the position diagramed in figure 7, the frontier squares for White are c4 and h4. The next step is to determine, for each of these frontier squares, the set of squares under Black's influence which, if the Black King were located on that square, would prevent the White King from moving from the frontier square to any of the contested squares or to any of Black's squares. For the frontier square at c4, the Black King would have to be at either a6 or b6 to prevent the White King from penetrating to b5. For the frontier square at h4, the Black King would have to be at g6 or h6 to prevent penetration by the White King. (Note that the Black King could not legally be at h5 if the White King were at h4.) These defense squares for Black can be determined by the machine by placing the White King on the frontier square and conducting a shallow tree search with White to move first and determining

empirically which locations for the Black King successfully repel the invader.

The next step in this process is to determine the shortest distance between each pair of frontier squares. For the present position, there are only two frontier squares and thus one minimal distance. Five King moves are required to travel between the two frontier squares. If Black is to be successful in defending, the Black King must be able to move from a defense square for h4 to a defense square for c4 in the same number or in fewer moves than it takes the White King to travel between the two frontier squares.

For this reason, each square in Black's defense set for c4 must be five or fewer moves from one of the defense squares for h4. Also, each square in the defense set for h4 must be five or fewer moves from one of the defense squares for c4. This requirement places a further restriction on those squares which satisfy the necessary defense conditions. One will note that a6 is six moves from the nearest square in the defense set for h4. Also, h6 is six moves from the nearest square in the defense set for c4. Therefore, the true defense set for c4 contains only b6 (a6 will not suffice). The true defense set for h4 contains only g6 (h6 will not suffice). Thus, we have determined that when the White King is on c4 and has the move, there is one, and only one, coordinate square for the Black King (b6). If the White King is on h4 and has the move, there is one, and only one, coordinate square for the Black King (g6).

The next step is to generalize this analysis to squares in White's territory which are immediately adjacent to the frontier squares. In this case, squares b3, c3, d3, g3 and h3. The square at b3 is one King move from the frontier square at c4 and six moves from the frontier square at h4. If the White King is at b3, therefore, the Black King must be on a square which is simultaneously one move from b6 and six or fewer moves from g6. The squares which satisfy this condition (ie: the coordinate squares for b3) are a6, a7, b7, and c7. This same set of calculations can be made for the other adjacent squares. The coordinate squares for c3 are b7 and c7. For d3, there is only one coordinate square, namely c7. Since the White King can move directly from c3 to d3 and Black must move to c7, and only c7, to maintain his defense, it is not possible for him to be on c7 when the White King is on c3. If he were, he would not be able to move when White moved from c3 to d3 and still satisfy the defense requirements. For this reason, only square b7 is sufficient for Black when White is on c3. In addition, since b3 is adjacent to c3, the coordinate square for c3 is not avail-

able for b3. Thus the set for b3 is further restricted to a6, a7 and c7.

If we examine g3, we will discover that it is one move from the frontier square at h4 and four moves from the frontier square at c4. This implies that the Black King must be on a square which is one move from b6 and four or fewer moves from g6. There are only two squares which satisfy this requirement, namely, f6 and f7. Therefore we can conclude that no square other than f6 or f7 will serve as a coordinate for g3. When we examine h3, we will find that there are three potential coordinate squares: f6, f7 and g7. Since this set shares f6 and f7 with the defense squares for g3, further restrictions are implied. It is not possible for the same square to serve as a coordinate square for two adjacent squares since it is not possible for Black to pass when it is his turn to move. Therefore if f6 is assigned to h3, then f7 *must* be assigned to g3. If f7 is assigned to h3, then f6 *must* be assigned to g3.

The next step in this process is to determine the set of coordinate squares for each square on the minimum pathway(s) between the two frontier squares for which the coordinate squares have not yet been determined. The new squares are e2, e3, f2 and f3. By following the same analysis as before, we can determine that the coordinate squares for e2 and e3 are d7 and d8. The coordinate squares for f2 and f3 are e7 and e8. Because of the adjacency restrictions, the assignment of one of these values automatically restricts the other square to the remaining value.

The results of our coordinate square analysis are summarized in table 1. When it is Black's turn to move and White has moved to one of the squares listed in the table, Black must be able to move to a coordinate square. For this reason, the evaluation function for the machine can be modified to give

Square of the White King	Coordinate Squares for the Black King
b3	a6, a7, c7
c3	b7
c4	b6
d3	c7
e2	d7, d8
e3	d7, d8
f2	e7, e8
f3	e7, e8
g3	f6, f7
h3	f6, f7, g7
h4	g6

Table 1: Results of the coordinate square analysis for the position of figure 7. Shown are the potential squares for the Black King which defend against the White King's threats when it is White's turn to move.

a bonus of 20 points to White for any terminal position in the look-ahead tree where it is Black's turn to move and the Black King is more than one move from a necessary coordinate square. If it is White's turn to move, a 20 point bonus will be awarded to any terminal position in the look-ahead tree where Black is not located on a necessary coordinate square.

Let us consider how this in combination with the White King attack map (figure 8) will affect the outcome of the look-ahead search. The machine will try to find a pathway to squares c3 or d3 because their attack value of 7 is higher than any of the surrounding squares. Even better would be a pathway to c4, since its attack value of 10 is larger than 7. In each of these cases, the machine will also try to satisfy the condition that Black cannot be on a proper coordinate square when the White King reaches c3, d3, or c4 so that the additional 20 point bonus is also earned. In attempting to do this, it will find that if the White King moves from a1 to either a2 or b2 on his first move, the 20 point bonus will be lost forever. The reason is that either of these moves allows the Black King to coordinate and, because of the minimax strategy, the tree search will always assume replies for Black which maintain this coordination. If the White King's first move is to square b1, the Black King cannot coordinate and the 20 point bonus *will still* be available at some of the terminal positions in the tree. It is not surprising, therefore, to find Reuben Fine advising that K-N1 is the only move for White which preserves the win.

In order for the machine to find this move, assuming that both the attack map and the coordinate squares information are incorporated in the evaluation function, a search of nine plies is required. This is a tremendous improvement over the 26 ply search required by the unmodified program. In order to actually win a Black pawn, the White King must move to c3 or c4 with Black not in coordination and make a 13 ply look-ahead search. If the White King moves to d3 with Black not in coordination, an 11 ply search will suffice. In order to prevent a draw, White will avoid repeating identical positions and thus will eventually travel to e3. From this vantage point, the win of a pawn can be visualized with a 9 ply search. Therefore, the problem could be solved by the machine if it searched to a depth of nine plies for each move calculation. With a program such as Chess 4.5, a 9 ply search for this position can be conducted in less than two minutes on even a medium power computer.

The procedures which we have described

are applicable to a wide range of end game positions. The coordinate squares analysis demonstrates that even highly complex end game positions are manageable when the full-width search employs a sufficiently knowledgeable evaluation function. Although the examples we have discussed encompass only a few types of chess positions, we hope that the reader will envision the power which is potentially available when the evaluation function is modified to incorporate relevant chess knowledge. The implementation of this approach on a broad scale should eventually produce chess programs which can be run on medium power machines and still compete on equal terms with strong human players.

Quiescence

Another important area for the application of chess knowledge is the problem of *quiescence*. It is essential that the static evaluation function not be applied to a turbulent position. If the next move has the potential to produce a major perturbation of the situation, the evaluation which is rendered will not be accurate. For example, it makes little sense to apply a static evaluation function in the middle of a piece exchange or when one of the Kings is in check. In each case, the judgment which is rendered will not be reliable. For this reason Chess 4.5 presently goes beyond the predetermined search depth at "terminal" positions where a capture might be profitable for the side whose turn it is to move, where certain types of checking moves are possible, or where a pawn is on the seventh rank. This extended search facility is called the quiescence search, and its major objective is to produce reasonably static positions for which the evaluation function can provide accurate assessments.

A weakness of this present implementation is that the definition of a turbulent position is much too narrow. There are many situations in addition to capture threats, checks on the King, and pawn promotion threats which are clearly turbulent. Larry Harris has characterized some of these in chapter 7 of *Chess Skill in Man and Machine*. Harris includes in this category positions which involve a pawn lever, a back rank mate threat, or sacrifice potential. The interested reader can consult Harris' chapter for operational definitions of these patterns. It is essential to note that these and other important patterns are not easily detected. In each case, a fairly sophisticated pattern analysis capability is required. A reasonable

goal for improving the present forward search chess programs would be the development of an efficient procedure for detecting potential sources of turbulence. The central objective would be to use this information as one of the decision criteria for terminating search at a node. If the position is not quiescence in respect to a potential perturbation which has been detected, the look-ahead process should be continued.

For example, during the opening when the machine leaves its library with information that the control of a particular square is an important objective, the decisions about search termination can consider whether the position is quiescent in respect to perturbations which might influence control of the key square. Another example of this idea involves the end game. If the preliminary analysis indicates that a particular pawn should be an attack target, the decision for search termination should consider whether each position is quiescent with respect to this goal. Positions at the predetermined depth level will be evaluated only if all potential attackers are more than two moves away from the target. When one or more attackers are close to the goal, the search process will be continued to determine if capture is feasible. This modification of the search process introduces a goal directed selective search at the terminal positions of the full-width tree. The addition of several extra plies of search at relevant nodes in the tree can mean the difference between finding and just missing an important continuation. This type of facility is difficult to implement and difficult to control properly, but the potential gains are such that the effort is worthwhile.

Establishing Appropriate Goals

In order to implement this goal direction feature in the evaluation function and quiescence search, it is necessary to recognize that a goal which may be of paramount importance at the base node of the look-ahead tree may no longer be relevant at some of the terminal nodes. Intervening moves may accomplish the necessary goal or may alter the situation such that it is no longer possible. In these cases, the conditional evaluation function would be directed at an inappropriate goal. One way to deal with this problem would be to select goals which were both general and long range. In this case, they should continue to be relevant at the terminal nodes of the look-ahead tree. Unfortunately, this is a fairly severe limitation on the goal directed search and is

therefore not desirable. A second approach would be to apply pattern analysis at each terminal node instead of at the base node only. In this case, the goals which were selected would always be relevant to the position. This procedure would be very time-consuming, since feature analysis is a complex process. The essential aspect of the problem is a time relevance trade-off in which a guarantee that relevant goals are being pursued requires a heavy investment in additional computing time. The third and most reasonable approach would be to designate which features of the position are crucial to each particular goal and to incrementally update our goals (and thus the evaluation function and the decision rules for the quiescence search) whenever these features change. This is a highly sophisticated approach which would be difficult to implement.

Conclusion

Let us summarize our conclusions and relate them to the world of personal computing. We have attempted to argue that a full-width search strategy is feasible with a small computer, and that ultimately this approach will produce better chess than a selective search strategy. For this plan to be successful, it is necessary to employ software and hardware suited to the task. The software must incorporate recent improvements in tree searching strategy (ie: α-β pruning, the capture and killer heuristics, iterative searching, staged move generation, incremental updating, serial evaluation and transposition analysis) as well as other refinements such as conditional evaluations which provide goal direction to the search process.

On the hardware side, it is necessary to have a reasonably powerful system. Although there have been a number of recent efforts to program microprocessor systems to play chess, the games which have resulted have not been comparable to those played by established large system programs. Although it is quite an accomplishment to produce even rudimentary chess from a microprocessor system, the level of play to date is not very encouraging. An example of this type of game appeared recently in March 1978 BYTE, "Microchess 1.5 versus Dark Horse," page 166.

The type of chess program described in this article requires reasonably powerful hardware in order to provide an interesting game. Because of the many operations requiring bit map manipulation, a 16 bit processor is much more desirable than an 8 bit processor. It is more efficient to represent

a set of 64 squares with four 16 bit words than with eight 8 bit words. With a need for computing power in mind, one might select a microprocessor system based on one of the new high-speed 16 bit processors such as the Zilog Z-8000 or the Intel 8086. In addition, this type of program will require quite a bit of memory. The program itself will require about 20 K bytes and the transposition table, if implemented, will need at least another 20 K bytes. If the programmer plans to add chess knowledge for conditional evaluations, a total of 64 K bytes is desirable. An opening library which is sufficient to keep a skilled opponent on his toes requires disk storage.

These considerations may dampen the enthusiasm of many would-be chess programmers. On the other hand, a realistic orientation at the start could save a great deal of grief along the way. When implemented on fairly sophisticated hardware, our demonstration chess program will usually provide a reasonable chess move after two or three minutes of computation. If more time is available (eg: selecting a move for a postal chess game by letting the machine "think" for several hours), a fairly respectable level of play can be anticipated. With future hardware improvements, this type of program may soon become reasonably competitive at tournament time limits, even on a personal computing system.■



REFERENCES

1. Berliner, H, *Chess as Problem Solving: The Development of a Tactics Analyzer,* unpublished doctoral thesis, Carnegie-Mellon University, Pittsburgh, 1974.
2. Church, K W, "Coordinate Squares: A Solution to Many Chess Pawn Endgames," undergraduate thesis, Massachusetts Institute of Technology, June 1978.
3. Church, R M, and Church, K W, "Plans, Goals, and Search Strategies for the Selection of a Move in Chess," Frey, P W (ed), *Chess Skill in Man and Machine*, Springer-Verlag, New York, 1977.
4. Fine, Reuben, *Basic Chess Endings*, David McKay Company Inc, New York, 1941.
5. Harris, L R, "The Heuristic Search: An Alternative to the Alpha-Beta Minimax Procedure," Frey, P W (ed), *Chess Skill in Man and Machine*, Springer-Verlag, New York, 1977.
6. Newborn, M, "PEASANT: An Endgame Program for Kings and Pawns," *Chess Skill in Man and Machine,* Frey, P W (ed), Springer-Verlag, New York, 1977.
7. Shannon, C E, "Programming a Computer for Playing Chess," *Philosophical Magazine,* volume 41, 1950, pages 256 thru 275.
8. Zobrist, A L, and Carlson, F R Jr, "An Advice-Taking Chess Computer," *Scientific American*, volume 228, number 12, June 1973, pages 92 thru 105.

Languages Forum

IPS, An Unorthodox High Level Language

About the Author and IPS

Karl Meinzer is a research physicist at the University of Marburg, in West Germany. He is a member of a very small, worldwide group of amateur satellite builders, who find an organizational home in the AMSAT project and its ties to the amateur radio community as well as various space agencies. Karl is the principal system designer for the high orbit "OSCAR" amateur radio satellites soon to be launched. His functions include the mechanical design, RF system design, computer system design and software design of the vehicle. IPS was developed in order to provide the AMSAT organization worldwide with machine independent OSCAR ground station software to be run on the typical project member's computer. The spaceborne version of IPS will run on an RCA 1802 processor in orbit, with 4 K (hopefully 16 K if technology becomes available) of on board memory and a radio communications bootstrapping method for loading from the ground control station. After the satellite is launched, IPS may indeed be one of the highest flying high level languages around.

Dr Karl Meinzer
Hoehenweg 38
355 Marburg 1
WEST GERMANY

It seems to me that the general confusion and lack of orientation regarding the best high level language for microcomputers is caused by the combination of two problems:

1. There is a marked lack of stated objectives regarding such a language.
2. It is tacitly assumed that the ease with which a feature can be added to a language is enough justification for adding the feature.

Let us look at the first point. In my experience, the term high level language has nearly the same meaning as "problem oriented" language. But oriented to what problems? It may help to classify computer applications into three large areas:

- *Mathematical problem solving:* Here the amount of data to be processed is usually small, but the data are subjected to relatively complicated numerical operations requiring high accuracy and diversity.
- *Commercial data processing:* Here large amounts of data are subjected to relatively simple procedures. Appropriate file handling techniques are essential.
- *Engineering applications in the wider sense:* By this I mean process control, systems programming, games, AI, robotics and any processes interacting with the real world. Programs in this area are characterized by moderate amounts of data and accuracy requirements. However, these programs usually have rather complicated flow of control.

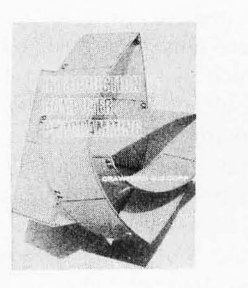
I think that much of personal computing would fall into the third class. On the other hand, the programming languages advocated so far were really created for the first two groups. To complicate things further, a good programming language not only has to be optimum for the intended problem area, but it also has to appeal to the humans using it. After all, the purpose of a programming language is to be a tool of communication between the human way of understanding a problem and the unambiguous way a computer needs the problem to be explained. Here of course many different opinions are to be expected, particularly because there is a positive feedback mechanism. By using a given language, one adapts to its idiosyncrasies and then, in all truth, can claim the "superiority" of that language. Nevertheless, I feel there are some language properties that would hardly be considered controversial. They can be best expressed by some buzzwords: top down design, structured programming, modularity and good self-documentation. These properties are particularly important in problem areas where the intelligence of the program resides in the flow of control.

The second problem may be explained by having a look at the competitive environment in which languages are created. If a company wants to express the superiority of their product, what better way than to claim to have more language features than the other guy. With this approach to computer languages lots of problems are created. The added features not only require additional implementation effort and an increase in the amount of hardware required (ie: memory), but they also make the language mentally less manageable, both from the

A Bit of the BASIC

———**Computer Resource Book—Algebra** by Thomas A Dwyer and Margot Critchfield is an exciting new way to learn about algebra and the interesting things you can do with it using a computer. The book uses the BASIC language, and flowcharts are used throughout to show the structure of programs. There are 60 applications programs including straight line graphs, polynomial equations, a space probe navigator, temperature profiles, computer generated animation, the ultramatic root finder, random number generation and many more. Although it is particularly suitable for students, just about everyone will find some intriguing and easy to use applications in this entertaining book. $5.20.





———**Introduction to Computer Programming** by Rudd A Crawford Jr and David H Copp. Here is an excellent way to learn about the general aspects of computer programming. **Introduction to Computer Programming** makes use of a hypothetical computer model and set of assembly language instructions designed to help the beginner see what goes on in computer programs. The emphasis throughout is on general principles; such concepts as loops, decisions, flowcharts and IO routines are covered in detail. The book also provides many example problems and prompts the reader by posing several quiz questions. Anyone who masters its contents will have a solid foundation for the study of practical assembly and high level languages. It is especially recommended for students, but just about everyone new to the subject should profit from it. $4.80.

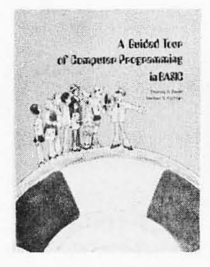
———**Basic BASIC** by James S Coan. If you're not already familiar with BASIC, James Coan's **Basic BASIC** is one of the best ways to learn about this popular computer language. BASIC (which stands for Beginner's All-purpose Symbolic Instruction Code) is easy to learn and easy to apply to many problems. **Basic BASIC** gives you step-by-step instructions for using a terminal, writing programs, using loops and lists, solving mathematical problems, understanding matrices and more. The book contains a wealth of illustrations and example programs, and is suitable for beginners at many different levels. It makes a fine reference for the experienced programmer, too. $8.95.

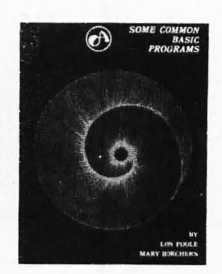




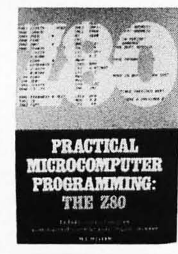
———**Advanced BASIC** by James S Coan. **Advanced BASIC** is the companion volume to James Coan's **Basic BASIC.** In this book you'll learn about some of the more advanced techniques for programming in BASIC, including string manipulation, the use of files, plotting on a terminal, simulation and games, advanced mathematical applications and more. Many useful algorithms are covered, including some clever sorting techniques designed to reduce program execution time. As with **Basic BASIC,** there are many illustrative example programs included. BASIC doesn't have to be basic with **Advanced BASIC!** $8.95.

———**A Guided Tour of Computer Programming in BASIC** by Thomas A Dwyer and Michael S Kaufman. Colorful graphics abound in this lively introduction to the BASIC language. The authors have tried to present a rigorous, yet entertaining approach to the subject. Written for the novice, **A Guided Tour** begins with a section on how to recognize a computer, followed by some tips on working at a terminal. By the end of the book readers are writing their own programs and solving elementary problems in finance and business. The emphasis throughout is on learning by doing. Anyone interested in computer programming should benefit from **A Guided Tour of Computer Programming in BASIC.** $5.20.





———**Some Common BASIC Programs** by Lon Poole and Mary Borchers, published by Adam Osborne and Associates. Here's a single source for all those hard to find mathematics programs! **Some Common BASIC Programs** combines a diversity of practical algorithms in one book: matrix multiplication, regression analysis, principal on a loan, integration by Simpson's rule, roots of equations, operations on two vectors, chi-square test, check writer, geometric mean and variation, coordinate conversion and a function plotting algorithm. These are just some of the many programs included. For only $8.50 you can buy the kind of programs previously available only as part of software math package systems for large scale computers. All the programs are written in a restricted BASIC suitable for most microcomputer BASIC packages, and have been tested and debugged by the authors. New, 2nd edition, only $8.50.



The Textbook. . .
18 Chapters (481 pages) of solid, accurate programming information, by Walter J Weller.

The Assembly Program (easily worth $100. by itself). . .
The full *source listing* of a combined editor/assembler which supports the language used. This language uses 8080 mnemonics, and is resident in less than 10K of RAM.

A Full Debugging Monitor. . .
Not a simple ROM monitor. Contains facilities for breakpointing, modification of pseudo registers and much more.
All this for the amazing price of $29.95.

And by the Way. . .
When the coupon in the book is filled out and returned, the object programs of the editor/assembler and debug are shipped FREE!

BITS T.M. inc Books to erase the impossible

POB 428, 25 Route 101 West
Peterborough, NH 03458

For your convenience in ordering, please use this page plus the order form on page 99. You may photocopy this page.
DIAL YOUR CHARGE CARD ORDERS TOLL FREE
(800) 258-5477.

master charge VISA

Circle 35 on inquiry card.



user's and the implementer's points of view. In my opinion, PL/I, despite all its virtues, is a good example of where this philosophy leads.

We Need Extensibility, Not Features

It is noteworthy that the need of features in languages arises in the first place because of their restrictive nature. In the traditional data processing environment with many independent users, this is justified by having to prevent inadvertent interactions. But in the personal area, with typically only one user per computer, one may take a much more liberal attitude. If one user bombs, nobody else suffers. So instead of packing a language with features, it makes much more sense to provide only those capabilities which will be required in, say, 70% or more of all programs. *In addition the language should contain means to extend itself.* Thus, it will be possible to add additional capabilities, if required. Another holy cow in high level languages is the use of algebraic notation. Because it is relatively easy to implement, its desirability is hardly ever investigated. Admittedly, expressions in reverse Polish notation are somewhat less readable than the equivalent algebraic forms.

However, look what you gain by the explicit presence of a stack:

- Parameter passing between program modules becomes extremely simple.
- The problems regarding the initialization and range of validity of variables in blocks disappear. Local variables are kept on the stack without a name; only global variables are named.
- With variables there is a clear distinction between the "pot" and the "content of the pot." The explicit availability of addresses enables very fancy pointer calculations.
- Testing of program modules is extremely simple. By manually supplying parameters on the stack and observing the results, debugging is a snap.
- The language becomes completely procedural and the statements are executed strictly in the order in which they are written down. There is no guessing or experimental programming to learn what the compiler will do to your program.

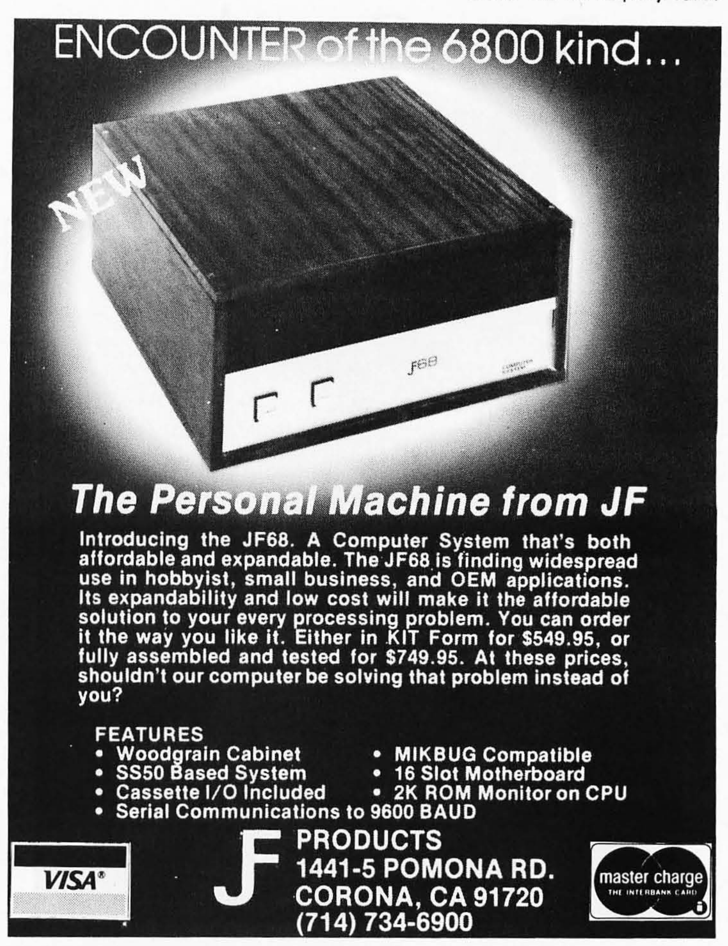
I hope that these points show that the decision — reverse Polish notation or algebraic — should not be a closed matter.

Another point often overlooked is the fact that the language itself is only part of the total acceptability of an approach. The amount and ease of handling are at least as important. A compiler requiring five passes before an executable code is produced hardly could claim to be very convenient. I feel it is highly desirable that a system should be interactive. This goal includes the requirement that a compiler should only require a single pass; incremental compilation with an interpretive mode should be available to simplify debugging. Usually this translates into the fact that objects need to be defined before they can be referenced, a small penalty for a much more transparent system. (Probably the wide acceptance of BASIC and APL is based on their interactive handling despite their mismatch to engineering problems.) Also, note that a fast single pass translation system eliminates the need for code relocatability. Instead of relocating binary objects, one simply translates the source to a new position in memory; there is no separate loading process.

If one takes the points above as an outline for a language design, it turns out that amazingly few degrees of freedom are still left. The main remaining work is selecting a suitable set of primitive functions, from which the language is built. This cannot be done arbitrarily, as a lot of feedback is necessary from actual problems utilizing the language. Only that way can one assure that the instructions are powerful and self-documenting in a reverse Polish notation environment. Most importantly, the language should have as few underlying rules as possible; and these rules should be applied without exception.

An important question relates to the optimum implementation technique to be used. Essentially three approaches are possible. BASIC and APL are generally realized as interpreters. Classical interpreters are rather slow because the program has to be analyzed at execution time. The other approach, the compilers, produces machine code which executes without syntax and semantics analyzing overhead; although a good optimizing compiler usually is bulky and requires large amounts of memory. Otherwise, the code produced is less than optimal if the compiler is simple.

A third approach tries to combine the best of both techniques. It uses a set of high level instructions defined by code routines. A compiler produces a pseudocode, treating these code routines as the instruction set of a virtual high level machine. Since these





instructions (usually addresses to primitive code) need very little overhead to interpret, such a system is only 2 or 3 times slower than optimum assembler produced code. Assuming that this is acceptable, the technique results in an extremely compact pseudocode requiring significantly less memory than the other approaches. Furthermore, the pseudocode itself is machine independent; only the primitive routines present an interface to the underlying processor. This greatly facilitates program portability, since transcribing such a system to a new processor entails rewriting on the order of 1000 bytes of code (600 in the example which I have implemented). Of course the largest part of such a system is written in the language that is to be implemented.

My interest in computing stems from my involvement with the AMSAT Phase III satellite, a space project for the amateur radio service. For the on board computer as well as for the ground stations I wanted to design a system that met the criteria above and would allow total portability of the programs among the numerous different computers used by radio amateurs worldwide. Naturally the first thing to do, if one tackles a project of this sort, is to study the literature on the subject. A language that meets most of the points just mentioned has been published by CH Moore under the name of FORTH. Particularly, Moore's implementation technique is very ingenious. On the other hand, his user interaction mechanisms were too limited for our problem area. Thus, I rather took FORTH as a platform, from which I designed and implemented my own system which I named IPS (abbreviated from its German language name).

The hardware I had in mind when I designed IPS would typically consist of:

- An 8 bit processor. (The spacecraft computer uses the RCA COSMAC and most ground stations use the 8080 and some the 6502.)
- 16 K bytes of programmable memory. The IPS system itself requires a little less than 6 K bytes, so enough memory is left to the user.
- An ASCII keyboard and a 64 character by 16 line video display memory. The video memory makes possible much more transparent interactions than obtainable with a Teletype or a hard copy oriented system.
- A hardware input of a sort providing 20 ms timing signals. All timekeeping and scheduling is derived from this input.

- Only enough read only memory to provide a simple bootstrap loader. (The COSMAC based systems have no ROM at all due to a hardware load feature on the chip.)
- A cassette interface and one or two cassette recorders. Since all ground to spacecraft and spacecraft to ground communication takes place synchronously at 400 bits per second, the same rate is used for recording. Recordings have a fixed length of 512 bytes (half a TV screen) and may consist of ASCII characters or binary data. Synchronous in this context means that there are no start or stop bits, the data is self-clocking and a valid block is preceeded by a 31 bit long sync vector.

I want to give you now a cursory description of the language. In the context of this discussion, unfortunately, it will be impossible to give you a detailed introduction. Rather, I intend to familiarize you sufficiently for being able to walk with me thru a typical program. I hope that this way you will be able to get some feeling of the nature of IPS. First let me describe how you communicate with IPS. The display screen has 16 lines. The eight lower lines are used for input to the system, either manually or by tape. The upper eight lines are used by IPS for answers. The basic IPS quantity is the 16 bit signed integer. By typing 125, for example, this number is put on the stack and displayed in the first display line. Keying −20 now, you get the display 125 −20. Now keying * will result in −2500 being left on the stack. An arbitrary number of numbers may be held on the stack; operations always refer to the last entries on the stack regardless of the total number of numbers on it.

Besides the normal algebraic operators logical operators are also available to allow bit manipulations. Boolean operations use these operators as well. (Only the effect on the least significant bit is utilized.) Another class of operations allows one to manipulate the order of items on the stack, like DUP duplicating an entry or VERT, which interchanges the two top entries. [Note: *In present versions of IPS, German (the author's natural language) inspired the symbols of primitives. Thus, VERT is from the German word Vertauschen.]*

Numbers may be stored permanently with a name attached to them. They can either be defined as constants (KON), as variables (VAR) or ar arrays (FELD). Calling a constant by mentioning its name pushes its value on the stack. Calling a vari-

Listing 1: The complete text of an IPS program's source. The program used as an example is the software which converts ASCII to Selectric codes, formats a text file and prints that file on a Selectric typewriter. Please note that the IPS primitives used in this listing are based on the German version of the system; equivalent English language names of the primitives are often noted in the text of the article as various features are explained.

```
( IBM SCHREIBMASCHINENSTEUERUNG VOM 2 . 11. 77 )
0 KON CBA
HIER 2 + ? CBA ! HIER 68 + $H !
CBA    VAR CHARP
0      VAR LE
#1000  VAR SBP
#1000  VAR IBP
0      VAR BL
CODE REMOVE 14 LD IM #C PHI 0 LD IM #C PLO 8 LD IM #A PLO
     BEGIN  #D I/O LD MX #20 AND IM ( BELEGT? )
     D=0 Y? #C DEC #C GHI              ( 140 US LOOP )
    VERT D=0 END TH #FF LD IM #A STR 3 I/O #A DEC 4 I/O NEXT

CODE H/TRANSMIT 1 LD IM #D PLO BEGIN 8 LD IM #A PLO PS INC PS LD
A #FF XOR IM  #A STR 100 LD IM #C PHI 0 LD IM #C PLO #A INC
      BEGIN #D I/O LD MX #30 AND IM  ( FREI? )
      D=0 NOT Y? #C DEC  #C GHI
     VERT D=0 END NEXT
             TH #A DEC #D GLO D=0 Y? 3 I/O
                                     N: 4 I/O
                                     TH NEXT
CODE L/TRANSMIT 0 LD IM #D PLO 0 END
CODE MSTATUS PS ->X PS DEC #D I/O #A ->X
          PS DEC 0 LD IM PS STR NEXT
0      VAR LV
0      VAR S
0      VAR EINFL
4   FELD SC          #1020 #400C SC 2 !FK
18  FELD 3ZEICHEN
128 FELD T1

: INCR DUP @ 1 + VERT ! ;
: ADJ DUP @ #3FF UND #1000 ODER VERT ! ;
: LIES LADEFLAGGE @B NICHT DUP LV @ UND
      JA? IBP @ 512 + IBP ! IBP ADJ   S INCR 0 LV !
      DANN
      EINFL @ S @ 2 < UND UND ( LADEFLAGGE NICHT )
      JA? IBP @ DUP #200 + $LOAD 1 LV !
      DANN ;

: NORMALZEICHEN DUP MSTATUS EXO #80 UND =0
             JA? DUP #80 UND >0 ( DREHE KOPF ) JA?    1
                                              NEIN:  2
                                              DANN
                 H/TRANSMIT 0 4 JE NUN REMOVE 0 6 JE NUN
             DANN L/TRANSMIT CHARP INCR REMOVE ;


: SONDERBEH #20 - DUP 4 < JA? DUP 2 = JA? 1 LE !
                                DANN
                          SC + @B H/TRANSMIT CHARP INCR
                          REMOVE
                      NEIN: CHARP @ 2 - CHARP !
                            4 - DUP DUP + + 3ZEICHEN +
                            CHARP @ 3 >>>
                      DANN ;
```

able or an array pushes its address on the stack; in order to get its value, the operator @ is used. It interchanges an address on the stack with the content of the address. To store a value into an address, the operator ! is used. It expects a value and an address on the stack, and by performing the storage, removes both. There are other similar operators like @B or !B, which perform like operations on single bytes.

Program modules are created by a colon followed by the name of the module. Then the actions are typed in as if the computer were to perform the actions immediately. A module is closed by a semicolon. Later on this module may be executed by typing its name or its execution may be put into another module simply by writing its name.

To control the flow of the program, IPS has some control words consistent with structured programming. The words JA? , NEIN: DANN (In English, respectively: YES? , NO: , THEN) are pretty self-explanatory. They roughly correspond to the IF THEN ELSE construct of other languages, only, to be consistent with reverse Polish notation, the test action precedes the JA? . There are three loop constructs available. The JE . . . NUN (in English, EACH . . . NOW) construct expects two numbers, a loop start index, and a loop limit. The loop is executed for each consecutive value, until the index exceeds the limit. If initially the limit is smaller than the start, the loop is not executed at all. This construct may also employ an increment different from one. The two other two loop constructs are intended for loops with an initially unknown number of repetitions. The ANFANG . . . ENDE? (in English, START . . . END?) structure is functionally equivalent to the DO UNTIL of PASCAL with the test at the end of the loop. The ANFANG . . . JA? . . . DANN/NOCHMAL (in English, START . . . YES? . . . THEN/AGAIN) structure corresponds to DO WHILE with the test at the beginning of the loop. Note that these constructs completely eliminate the need of GOTOs and labels.

Now I want to take you on a little walk thru a program. Its purpose is to read ASCII text recorded on tape cassettes and print them on my IBM Selectric typewriter. (As evidence of the practicality of IPS, this article's manuscript was printed by this program.) As mentioned, a tape block contains 512 characters (eight lines) without any control characters. The program, besides reformating the ASCII characters to the Selectric code, has to insert this control information. The program maintains a cyclic buffer of two blocks; S is a variable

Circle 71 on inquiry card.

Radio Shack Computer Users

TRS-80 monthly newsletter

The largest publication devoted to the TRS-80 System

- **Business**
- **Personal Finance**
- **Practical Applications**
- **Gambling—Games**
- **Latest RADIO SHACK Developments**
- **Software Exchange**
- **Market Place**
- **Questions and Answers**
- **Program Printouts**
- **. . . and more**

MAJOR PROGRAMS PUBLISHED MONTHLY • INCOME TAX PROGRAM – LONG AND SHORT FORMS • INVENTORY CONTROL • EXTENSIVE MAILING LIST AND FILE PROGRAM • PAYROLL • STOCK SELECTION • PICKING WINNING HORSES • RENUMBER PROGRAM LINES • CHESS • CHECKERS • FINANCIAL APPLICATIONS PACKAGE • PERSONAL FINANCE PACKAGE • GRAPHICS • STATISTICS • MATHEMATICS • EDUCATION . . . AND MORE

$24. Per Year





Box 149RB, New City, New York 10956 (914) 425-1535

Send for FREE Software Catalogue (Including listings of hundreds of TRS programs available on cassette and diskette).

Circle 3 on inquiry card.



READY-TO-RUN SOFTWARE

TRS-80 • NORTH STAR • APPLE

INTEGRATED BUSINESS SYSTEM $175.00
A/P; A/R; Gen Led; Pay; Bus Stat; Letter Writer. Comes "Ready-to-Run" with easy-to-follow instructions. Requires 2 disc drives. Specify: TRS-80, NORTH STAR, or APPLE II (must have Applesoft).

TUTORIALS $40.00
Computer Aided Instruction, teaches the idiosyncracies of your version of disk basic. Specify: TRS-80, NORTH STAR, APPLE II (Applesoft).

HOMEMAKER I $50.00
Includes: Appointment Minder, Shopping List, Menu Planner, Diet Planner. Specify: TRS-80, NORTH STAR, APPLE II.

EDUCATOR I $50.00
Includes: Math Tutor, Grading System, Hangmath, Spelling, IQ Test, Games. Specify: TRS-80, NORTH STAR, APPLE II.

WORD PROCESSOR $35.00
Comes complete with simple, easy-to-follow instructions.

ALL SOFTWARE PACKAGES ARE WRITTEN IN BASIC ON DISKETTE ONLY

Send for Free Catalog

Packages containing Program Listings only are available.

(714) 774-1270



AJA SOFTWARE
P.O. Box 2528 • Orange, CA 92669

APPLE II USERS
WE'VE GOT SOFTWARE!

We have the best disk file management system in the market. Use it for your mailing list, accounts receivable, payables, personnel records, etc.

In addition, we offer not only the most fascinating basic programming course for people with little or no programming experience, but also commercial software for North Star Horizon Computers and Ohio Scientific Computers, and over one hundred programs for Apple II Computers.

The file management system is available for $60.00 and the programming course for $30.00.

Please specify tape or disk version.

Check, money orders and credit cards accepted.
Florida residents add 4% tax.

TRANS-DATA CORPORATION
161 Almeria Avenue
Coral Gables, FL 33134
(305) 576-7666

Send $1.00 for software catalog or call our toll free number (800) 327-8455.

Circle 378 on inquiry card.

DATAFACS

Software now for:
Micropolis and **NorthStar**

Good Software is Essential for an Efficient Business.
Now, the finest in software at a price you can afford.
We are definitely the affordables!

PROGRAMS CURRENTLY AVAILABLE:

INVENTORY PROGRAM PACKAGE

- Inventory Addition • Update
- Analysis (Current month) • Analysis (Year to date)
- Search • Vendor File

Also generates 7 additional user reports and much, much more.

— Cost $175.00 —

MAILING LIST PROGRAM (Single Drive)

*Print formating in 80 or 132 columns.

— Cost $20.00 —

The above programs are designed to run under the following:

*CP/M in CBASIC / *CP/M in BASIC E } Micropolis and NorthStar (2 drives required) also IMSAI IMDOS

(2) 8″ disc IBM 3740 Format required

EXTENSIVE CONTRACTORS ESTIMATING PROGRAM

Literature available upon receipt of your letterhead

Currently Under Development:

- Billing
- Payroll
- General ledger

} Available early in '79 in new CBASIC-2

Specializing in custom software for the AM-100**

Datafacs System Inc. Eastern Division
2440 W. Catalpa, Chicago, Ill. 60625
Phone (312) 784-0300

*CP/M is a Trademark of Digital Research Corp.
**AM-100 is a Trademark of Alpha Microsystems.

Circle 81 on inquiry card.

Listing 1, continued:

```
: TYPLINE #0D ( CR/LF )  CHARP @ 1 + !B CBA CHARP ! 0 LE !
        ANFANG CHARP @ @B #7F UND T1 + @B
               DUP DUP #1F > VERT #30 < UND
               JA?   SONDERBEH
               NEIN: NORMALZEICHEN
               DANN LE @
        ENDE? ;

: SZEILE SBP @ CBA 64 >>> CBA 1 - CHARP !
       CBA DUP 63 + JE I @B #20 - >0
                      JA? I CHARP !
                      DANN
                   NUN TYPLINE ;


: SBLOCK 0 7 JE SZEILE   SBP @ 64 + SBP ! LIES
             MSTATUS 2 UND >0 ( STOP? )
             JA? 0 EINFL !
             DANN
          NUN SBP ADJ S @ 1 - S ! BL INCR ;


: SCHREIB EINFL @ JA? S @ >0
                   JA? SBLOCK BL @ 4 >
                       JA? 0 BL ! MSTATUS    ( START? )
                                  JA? 0 7 JE CBA 1 - CHARP !
                                             TYPLINE
                                         NUN
                                  NEIN: 0 EINFL !
                                  DANN
                       DANN
                   DANN
               NEIN: 0 BL ! MSTATUS #80 UND =0
                             JA?  ( TIEFSTELLUNG BEI AUS )
                                  2 H/TRANSMIT REMOVE DANN
                      MSTATUS ( START? ) JA? 1 EINFL !
                                        DANN
               DANN
      MSTATUS 2 UND >0
      JA? 0 EINFL !
      DANN LIES ;


( ENDE SCHREIBMASCHINE; SCHREIB IST EINGEHAENGT )


( CODETABELLE FUER IBM KOPF CORRESPONDENCE COURIER )
#2020   #2020   #2020   #2020   #2321   #2020   #2220   #2020
#2020   #2020   #2020   #2020   #2020   #2020   #2020   #2020
#B420   #87B7   #CF26   #58D7   #DE96   #4625   #0119   #C634
#7FCC   #3E37   #574F   #5E16   #071F   #FFC9   #BE27   #9928
#9C24   #9A82   #D2DB   #F9B8   #95C3   #93F0   #FCCA   #CCB2
#90D1   #C5DD   #BBF3   #84BD   #C0FA   #54F6   #7531   #8129
#1CD8   #1A02   #525B   #7938   #1543   #1370   #7C4A   #4C32
#1051   #455D   #3B73   #043D   #407A   #D476   #F5B1   #2049

   T1      64 !FK

#084F   #582B   #2B08   #0853   #2D2F   #2708   #082D   #6060
#2708

  3ZEICHEN 9 !FK     ( ENDE TABELLE )
```

indicating the number of blocks available for printing. S equals 0 means an empty buffer, S equals 2 is a full buffer.

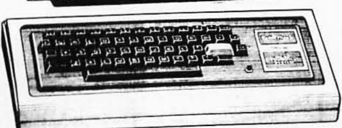
Complete details are found in listing 1. Turn first to the last routine SCHREIB (in English, WRITE) which is the main program executed periodically. (Because of the previously mentioned referencing restriction, programs have to be read backwards.) SCHREIB at first checks if EINFLAG (ONFLAG) is set; if so, it starts the printing actions. This action starts by checking if S is larger than 0. Only if this is the case is there a block in the buffer and the routine SBLOCK (WRITE BLOCK) is called. A blockcounter is maintained (BL). If it exceeds four, printing is stopped (PAGE FULL) unless the start switch is on. In this case the 0 7 JE . . NUN does eight linefeeds (endless paper assumed). The other parts of the main program are less important, they mainly make sure that at the end of printing the head is in lower case because if left in upper case the machine cannot be manually switched into lower case.

Now let us look at the routine SBLOCK (WRITE BLOCK). It prints 8 lines (0 7 JE . . NUN) by calling the routine SZEILE (WRITE LINE) and then updating the write buffer pointer SBP by 64. Also it checks for an emergency stop. The routine SZEILE (WRITE LINE) takes 64 characters at SBP and transports them into an auxiliary array CBA (>>> is an array transport in IPS). Then the auxiliary array is scanned in such a way that the pointer CHARP points at the last printing character. The loop index is supplied by I. Then TYPLINE is called. TYPLINE inserts a CR/LF after the last printing character and then uses the characters in the array CBA without the most significant bit (#7F AND) as index to table T1 to find the Selectric code. If it is between #1F and #30 a special treatment is necessary by calling SONDERBEH, otherwise NORMALZEICHEN (NORMAL CHARACTER) is called.

Let's look at SONDERBEH (SPECIAL CHARACTERS). Some actions of the machine are not performed by the printing ball, but by special magnets (blank, for example). In my hardware, these codes have to be output to a different port. Some ASCII characters are not available on the printing ball. These are simulated by two other characters overwritten on each other. The auxiliary array is accordingly manipulated. Both activities are done by SONDERBEH. The NORMALZEICHEN routine checks if the print ball is in the proper case position for the character to be printed. If not, it first executes a head



TRS-80 USERS: Make Two Bits Perform Like 16K.



If you missed the first two issues of **SoftSide**, you missed:

Cribbage A 16K you vs. computer game that's hard to beat. **Calculator** Just what it says. **End Zone** A 2-player 16K football game, from coin toss to 2-minute warning. **Pillbox** Computerized artillery for 2 players. **Shopping List** An aid for consumers Plus, **Troll's Gold, Octal to Hexadecimal Converter, Death Star and much more !**

A 1-year subscription will bring you about sixty top quality programs for $15.00 — or about 25 cents apiece. Think about it . . . how else can you get so many K bytes of programming out of two bits !

1 Year Subscription—$15
Check, money order or charge to VISA or Master Charge (send full address, signature, card numbers, exp. date with order, or telephone **603-673-5144**) to:
SoftSide Subscriptions
PO Box 68
Milford, NH 03055



NOBODY SELLS THE BEST FOR LESS!

COMPUTER LAB OF NEW JERSEY

	LIST PRICE	SPECIAL PRICE
Electronic Control Technology TT-8080 Computer Kit	440.00	375.00
PolyMorphic Poly 88 System 16	1,995.00	1,695.00
Sanyo 9" Video Monitor	220.00	165.00
Mullen Extender Board	35.00	29.00
Imsai S10 2-2 kit	156.00	132.00
Integral Data System IP-125 Printer	799.00	679.00
North Star Horizon II kit	1,999.00	1,699.00

SUBJECT TO AVAILABLE QUANTITIES. SHIPPING AND INSURANCE EXTRA.
SHIPPING FREE ON PREPAID ORDERS.

COMPUTER LAB OF NEW JERSEY

141 ROUTE 46
BUDD LAKE, NEW JERSEY 07828
(201) 691-1984



PSEUDO PROGRAM COUNTER
PSEUDO PROGRAM COUNTER +1
PSEUDO PROGRAM COUNTER AFTER CALL

n+1	n+2	n+3	n+4	n+5		M
INSTRUCTION	INSTRUCTION	CALL INSTRUCTION	ADDRESS M	INSTRUCTION N+5		INSTRUCTION

INVOKE "CALL"

```
EXECUTABLE CODE
OF INTERPRETER'S "CALL"
INSTRUCTION:
    PUSH PSEUDO-PROGRAM-COUNTER.
    PSEUDO-PROGRAM-COUNTER:=
        CONTENTS OF MEMORY AT
        PSEUDO-PROGRAM-COUNTER+1
    RETURN TO INTERPRETER
```

Figure 1: The temptation on the part of an interpreter writer is to implement direct execution as if the interpreter were simulating some form of traditional machine. Thus in this example, the operation code for a CALL to a subroutine is followed in the interpretive text by the address of the subroutine being called. This requires, for example, two entries in the string of interpretive text.

rotate action before printing. Each machine action is started by issuing the magnet code via H/TRANSMIT or L/TRANSMIT. These routines check the machine for "not busy" and then output the code to the magnets. The routine REMOVE does the opposite; if the machine acknowledges a code by "busy," the magnets are deenergized. Thus the necessary handshaking is established.

These interface routines are particularly noteworthy, because they are coded in assembler. IPS has an integral assembler; thus procedures dealing with special hardware or extremely time critical jobs can run at the maximum speed of the processor. This assembler also employs structured programming. The branch codes have been replaced by the Y? N: TH and BEGIN END mnemonics. The assembler, like the compiler, uses the stack to keep track of the addresses to be inserted into the branches. The routine LIES finally serves to read in the cassette blocks.

I hope this walk was not too tedious for you and was justified by giving you some insight into IPS programs and IPS-like languages. Let us have now a look at some selected topics regarding the implementation of IPS. These items maybe useful to you, if you want to tackle a similar project.

As mentioned earlier, the stack keeps 16 bit numbers. In addition to the normal operand stack, the design is simplified if there is a second stack, which is primarily used by the system to keep return addresses. This stack is available on a limited scale to the programmer as well.

A very useful trick in such a system is the use of "indirect execution." When implementing a simulated high level computer one is tempted to compile addresses of the appropriate code routines performing the instructions as shown in figure 1. This is a simple model of an "emulator" of some instruction set of a traditional computer.

If pointers are compiled that point to a descriptor which includes another pointer to an actual executable code, then all compiled instructions have the same format regardless of their nature (see figure 2). Indirect execution not only saves memory, it also simplifies the compiler because it needs no more checking the semantics of the instruction to be compiled. Thus in figure 2, a subroutine reference reduces to a pointer to the descriptor which begins the interpretive code of the subroutine. Thus, it boils down to having a descriptor at the head of each routine, defining its nature. This also makes the language naturally extensible since new descriptors may be defined anytime, and it eliminates the distinction between application routines and the primitive routines constituting the language.

In IPS the names along with the pointers to the pseudocode are kept in a hash table with a size for 512 entries. Experience shows that using this size in a 16 K system, memory and table saturate at about the same time. I preferred the hash table over a linked list because the compilation process is much faster and enables transcompilations for machines having no interaction facilities like the computer running the typewriter on which I printed this manuscript. (Names are separated from the program.)

Use of IPS in program development situations also requires the possibility of eliminating entries from memory. Generally a hash table would be unsatisfactory for this purpose. Nevertheless, since the program grows linearly in memory, the pointers to the routines in effect represent a secondary index by which the order of the entry and creation is available. Thus, pruning is no problem.

Reverse Polish notation languages can be essentially syntax free and by indirect execution the compiler can compile virtually blind, that is, without knowledge of what is being compiled. This is not exactly true, though. In fact, there are four different sorts of entries regarding the action the compiler is expected to take. This 2 bit

A Message to our Subscribers

From time to time we make the BYTE subscriber list available to other companies who wish to send our subscribers promotional material about their products. We take great care to screen these companies, choosing only those who are reputable, and whose products, services, or information we feel would be of interest to you. Direct mail is an efficient medium for presenting the latest personal computer goods and services to our subscribers.

Many BYTE subscribers appreciate this controlled use of our mailing list, and look forward to finding information of interest to them in the mail. Used are our subscribers' names and addresses only (no other information we may have is ever given).

While we believe the distribution of this information is of benefit to our subscribers, we firmly respect the wishes of any subscriber who does not want to receive such promotional literature. Should you wish to restrict the use of your name, simply send your request to BYTE Publications Inc, Attn: Circulation Department, 70 Main St, Peterborough NH 03458. Thank you.

information is kept in the name table.

Normal entries execute in the interpretative mode and compile in the entry mode. See table 1 for the other modes. INT is used for variable definitions or for the colon to prevent definitions within definitions. PRIOR, on the other hand, is used for the JA? NEIN: DANN and the semicolon. These entries are executed at compilation time to handle jump addresses on the stack or to reset the compile flag. At the beginning of compilation (:), a special number is placed on the stack. At the end of compilation (;), this number is checked. If it is wrong there has been a structuring error (such as DANN following a JA?) forgotten and the entry is rejected.

At this point I would like to point out that much of the readability of structured programs seems to come from a geometrical representation of the source text as exemplified by the indented writing. Hardly any language software analyzes this geometry, though, and it is in effect redundant to the semantics of statements.

There is another useful technique which I call "pseudointerrupts." On a stack computer programs are naturally reentrant; and recursive programming is possible. Thus, the inner interpreter can be designed to accept "interrupts" between actions specified by interpretive code. These interrupts interrupt a stack oriented computer simulated by the interpreter, so no state saving operations are necessary (except for the pseudoprogram counter, of course.) Machine interrupts may also be implemented independently when required. The 20 ms clock input assumed by IPS forces such a pseudointerrupt to keep time in a normal clock and to service four "stopwatches."

Engineering problems usually require a few programs running independently of each other. I solved this problem by providing

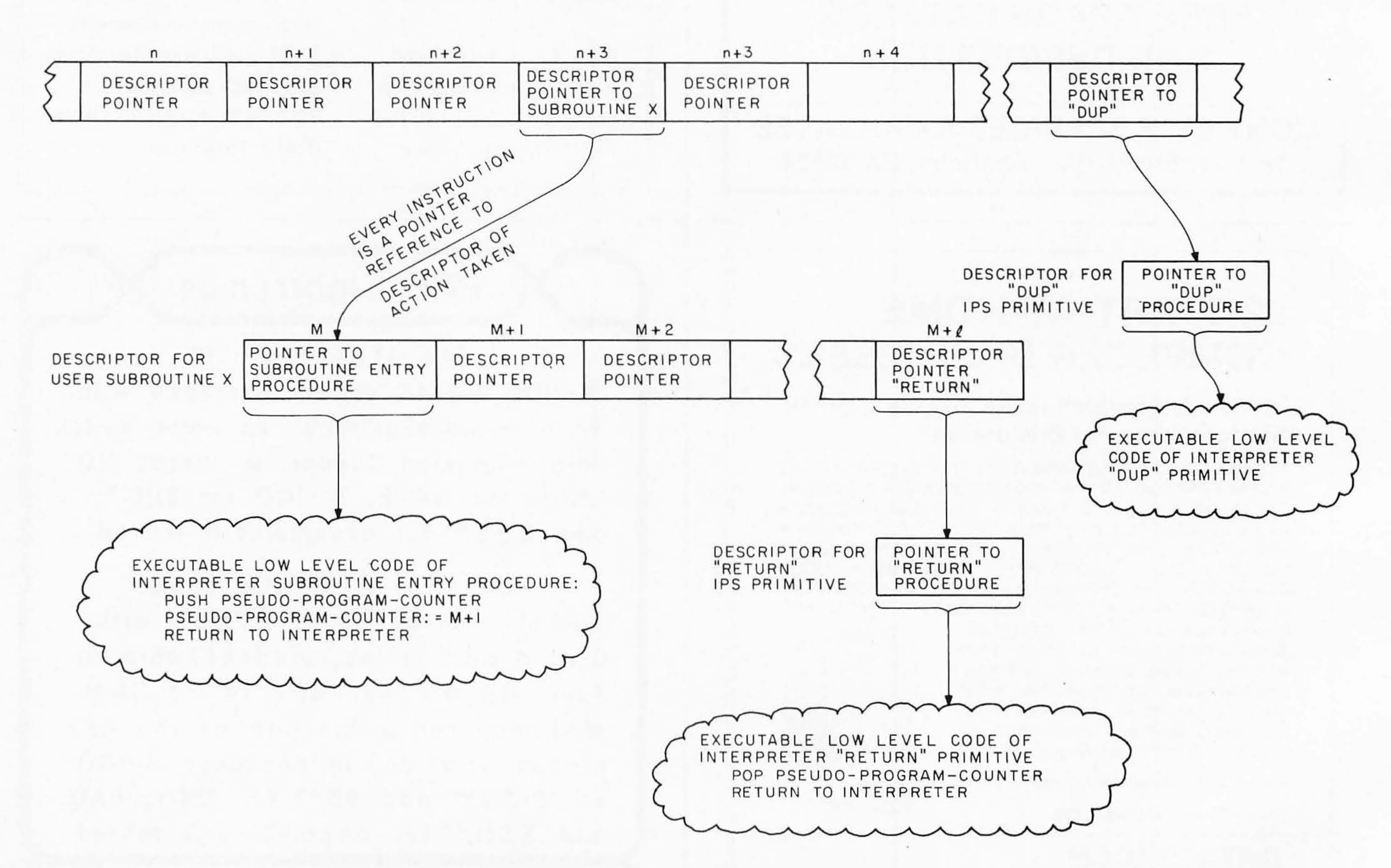


Figure 2: If a 2 level indirect form of interpretive execution is defined, then simplicity is achieved by making every token in the text of a user program a pointer reference to a descriptor of the operation. The descriptor in turn contains information such as the pointer to a low level routine to be executed upon reference to the descriptor during interpretation. Thus, a subroutine call which required two interpretive text entries in figure 1 is replaced by a simple pointer reference to the subroutine's descriptor. This descriptor in turn contains a reference to the low level interpreter routine which saves the program counter on the subroutine stack, then points to the first entry of the subroutine's interpretive text before resumption of interpretation. For simple built-in primitives of the language, the descriptor simply points to the low level executable routine of that primitive's operation, as shown for example in the case of RETURN and DUP. Note that the interpreter increments the pseudo program counter after each interpretation cycle.

a chain, that is, an array of programs executed in a cyclic fashion. Initially, this chain is only occupied by the compiler, while the other positions contain null entries. Programs may put other programs into this chain, or programs may remove themselves or other programs from this chain. It turns out that by this method, very general multiprogramming is possible without needing separate stacks for each chain member. The act of scheduling or descheduling a real time process is implemented by this chaining technique. Apart from the "chain in" and "chain out" operators two other operators are necessary. A suspend and a resume operator are provided to be able to wait for internal or external events before a program's execution is continued. Amazingly, the four chain operators eliminate the necessity of a centralized operating system without imposing undue restrictions.

Presently at AMSAT we have IPS versions for the RCA COSMAC and the 8080, and a 6502 version is in preparation. Also, there exists a special version for the COSMAC aboard the AMSAT satellite, performing all interactions by radio link. Except for the last they occupy a little less than 6 K bytes of memory, having everything resident including the text editor. The radio version is slightly smaller.

In mid 1976 the first IPS version was up and running. Since then I have done some fine tuning as a result of feedback from some ten scientific projects using IPS within the University of Marburg. If you feel that IPS could be useful to you or if you have some comments or suggestions, please do write me.

(The original paper which discusses such a machine interpreter is: "FORTH, a New Way to Program a Minicomputer," by C H Moore, Astron Astrophys Suppl 15, 497-511, 1974. In the years since the first FORTH paper, FORTH has become a registered trademark of Mr Moore's consulting firm FORTH Inc.)■

entry-type	compiler-mode interpreting	compiling
Normal (:)	execute	compile
INT	execute	error action
PRIOR	error action	execute
HPRI	execute	execute

Note: error action marks input-word with question mark and stops further processing of input.

Table 1: The modes of IPS operation (interpreting or compiling) and the action the compiler takes depending on the type of entry found. Normal refers to standard modules (:), variables and constants.

Clubs and Newsletters

Attention: Canadian Readers

We have heard from the West Coast Computer Society (POB 4476, Vancouver BC, Canada V6B 3Z8) regarding their club's meeting schedule. They meet the first Wednesday of every month at 8 PM, room 126 at BCIT, which is located in Burnaby, British Columbia.

TRS-80 Monthly Newsletter

The purpose of the *TRS-80 Monthly Newsletter* is to provide and exchange information related to the Radio Shack TRS-80 computer system. This 68 page newsletter contains programs related to business, personal finance, money management, practical applications, statistics, mathematics, gambling and games. Every month one major program (such as a complete personal finance package or a complete income tax program — short and long form) will be published. Subscriptions are $24 by first class mail ($40 outside of the US and Canada, air mail). Contact Mathematical Applications Service, POB 149, New City NY 10956.

Independent Ohio Scientific Users Group Formed

Owners, prospective users, OEMs and dealers of Ohio Scientific computers can share information, applications and software through an independent users group formed by the Newton Software Exchange, POB 518, Newton Center MA 02158. The Exchange was created to aid in the translation of microcomputer applications and systems programming from the sophisticated user and programmer to the naive user and operator. They are oriented and willing to work with programmers and OEMs who have good ideas and programs, but have difficulty expressing themselves to relatively unsophisticated end users. The OSI users group will encompass the full range of OSI products with special attention to the Challenger series. Annual dues are $5 and include a monthly newsletter.



Rhode Island Computer Club

RICH is a computer club in Rhode Island that will be meeting on the following dates: March 20, April 17 and May 15 1979, from 7 to 10 PM at Jabour's Electronic City, 345 Fountain St, Pawtucket RI. For more information, write POB 599, Bristol RI 02809 or call (401) 253-5450.

User Notes for 6502 Owners

KIM-1/6502 User Notes is getting a facelift and will now be called *User Notes: 6502*. It will continue as a bimonthly publication but will be expanded to 24 pages. In addition to the KIM, the *Notes* will support VIM and AIM systems as well as others. The new subscription rates will be $13 for six double issues mailed first class and to Canada and $19 for six double issues air mailed overseas. The new address is POB 33093, N Royalton OH 44133.

PACS Publishes Data Bus

The Data Bus is the monthly publication of the Philadelphia Area Computer Society (PACS). Its 16 pages contain a variety of information, including regular features such as meeting news, PET shop, editorial, classified and such articles as "Computers for Kids" and "Software/Courseware Exchange for UCSD Pascal." This organization was formed for the purpose of education and intercommunication of computer users in the greater Philadelphia area and to inform the general public about computer technology and its implications for the future. Dues are $10 per year for regular membership and $5 per year for students. Both memberships include a subscription to *The Data Bus* and participation in group purchase programs and club projects, as well as borrowing privileges in the society's software and literature libraries. PACS meetings are held in the LaSalle College science building. For more information, call the club's hotline at (215) 925-5264.

New York Amateur Computer Club

The New York Amateur Computer Club, which just celebrated its third birthday, is an organization formed to promote the free exchange of information about computers for personal use and to encourage fellowship among those interested in computing. General meetings are held once a month, normally on the second Thursday. Several user groups consisting of club members with specialized interests also meet on a monthly basis. Club dues are $10 per year which includes a well-compiled and informative newsletter. For more information about this group, write to the club at POB 106, Church St Station, New York NY 10007.

Microcomputer Owner's Society

Jim Brown of Elyria OH has notified us of the formation of the Microcomputer Owner's Society. They meet every fourth Sunday at 1 PM. For more specific information, contact Jim Brown, POB 474, Elyria OH 44035.

Compucolor-Intecolor Users Group

The Compucolor-Intecolor Users Group has announced that it will start making programs available to members without the donation of a program. Club members may now order programs from the club's library of over 300 programs for as little as $2 each, plus a disk and handling charge of $20 for the first disk and $15 for each subsequent disk ordered at the same time. Those donating an acceptable program will receive five or six programs back at no charge. Membership to the group is $25 or $10 with the submission of an acceptable article for publication in the group's bulletin or a program to the library. Those wishing to join should write to Compucolor-Intecolor Users Group, 5250 Van Nuys Blvd, Van Nuys CA 91401.

Tulsa Computer Society Sponsors Apple User Group

The Tulsa Computer Society (TCS) meets the last Tuesday of every month at 7:30 PM. The meeting place is the Tulsa Vocational-Technical School seminar room at 3420 E Memorial Dr. The Apple User Group meets the second Tuesday of the month at High Technology of Tulsa. All users of Apple II computers and interested individuals are invited to attend. Membership in TCS is $6 annually and includes a 1 year subscription to their newsletter, *The I/O Port*. Address all correspondence to The Tulsa Computer Society, POB 1133, Tulsa OK 74101.

Attention: Rochester NY Computerists

According to the most recent edition of *Memory Pages*, the official magazine of the Rochester Area Microcomputer Society (RAMS), the club is into its third year and has over 150 members with diverse backgrounds in the field of computing. RAMS is the local focus of personal computing in the Rochester area, as evidenced by the ACM and IEEE joining them for joint meetings on personal computing. It is the club's aim to provide its members with exposure to ideas and contact with individuals with similar interests. If you are interested, attend one of their meetings on the second Thursday of each month at 7:30 PM in room 1030, Gleason Building (#7 — Engineering), Rochester Institute of Technology, Rochester NY. The address for RAMS is POB D, Rochester NY 14609. ■

BYTE's Bits

Attention: ACM Members and Students of Computing History

The editors and editorial board of the *Annals of the History of Computing* are currently soliciting articles for this new quarterly periodical. The *Annals* will be published beginning in July 1979 by the American Federation of Information Processing Societies Inc (AFIPS). The publication will focus on the history of computing by including contributions from individuals who participated in, or witnessed, the events and decisions which have shaped the present computing environment.

In addition to formal articles, individuals are invited to submit short anecdotes or personal recollections, commentaries on collections of private letters considered historically relevant, descriptions of current historical projects, articles on the influence of societal factors on the development of computing, reports of significant successes and failures in the computing field, and annotated bibliographies of relevant publications. Coverage will center on events and developments that occurred at least 15 years prior to the date of publication.

Prospective authors interested in submitting articles should send five copies of their contributions to Bernard A Galler, editor in chief, *Annals of the History of Computing,* University of Michigan, College of Literature, Science and the Arts, 2522 LSA Building, Ann Arbor MI 48109. Articles should be between 5000 and 10,000 words. Longer articles may be considered for publication in two or more parts. All contributions should be typed, and diagrams should be carefully drawn and fully labelled. Glossy photographs may be included. All cited references are to appear at the end of the paper under the heading "Notes." The notes should be typed, numbered consecutively, and consist mainly of citations. All words to be italicized should be underlined.

All articles should be accompanied by a brief biographical sketch of approximately 50 words, an abstract of approximately 100 words, and a listing of appropriate *Computing Reviews* categories.■

TRS-80 Memory Expansion

Owners of Radio Shack TRS-80s, take note: if you are upgrading from 4 K to 16 K of user memory and have the work done at a Radio Shack repair center, you should ask that the old 4 K memory integrated circuits be returned to you, after the new 16 K memory devices are installed. There is no additional charge, but in most cases you don't get the old memories back unless you ask.

Those owners of TRS-80 systems who buy 16 K memory kits from sources other than Radio Shack should be aware of this precaution. It is likely that Radio Shack repair centers will charge more for repairs to TRS-80s that have been modified by the user than for repairs to machines untouched internally except by the hands of the technicians at said repair centers.■

Speakers Invited to Computing Festival

The Personal Computing Festival of the 1979 National Computer Conference will be at the Americana Hotel in New York City, June 5 through 7. The Festival will include three days of programs and special activities on all aspects of personal computing, with emphasis on the applications people have for their own computers. Individuals interested in presenting a paper, giving a talk, organizing a panel, or delivering a tutorial should send a letter of intent as soon as possible, but no later than February 1 1979, to Jay P Lucas, 3409 Saylor Pl, Alexandria VA 22304. The letter of intent should include an abstract and a brief biography. Papers presented during the festival program will be published. Potential authors will be mailed a festival author's kit, which contains instructions and necessary materials for preparing papers in camera ready format. Papers submitted for consideration must be received by March 15 1979, in the specified camera ready format. Papers will be reviewed and authors will be notified by May 1 1979 regarding acceptance. Session leaders should submit a brief abstract describing either the scope of the proposed session or the tentative title of the presentation by February 1 1979. The prospective organizer should submit a list of proposed participants, their affiliations, and a brief biography of each.■

On the Eight Queens Problem

When Terry Smith bypassed the permutations of digits method of "Solving the Eight Queens Problem" (October 1978 BYTE, page 122) because calculating the permutations of the

SURPLUS ELECTRONICS

ASCII ASCII ®

IBM SELECTRIC BASED I/O TERMINAL WITH ASCII CONVERSION INSTALLED $645.00

- Tape Drives • Cable • Cassette Drives • Wire • Power Supplies 12V15A, 12V25A, 5V35A Others, • Displays • Cabinets • XFMRS • Heat Sinks • Printers • Components

Many other items
Write for free catalog
WORLDWIDE ELECT. INC.
130 NORTHEASTERN BLVD.
NASHUA, N.H. 03060
Phone orders accepted using VISA or MC. Toll Free 1-800-258-1036
In N.H. 603-889-7661

Circle 395 on inquiry card.

COMPUTER MART of NEW HAMPSHIRE, Inc.

* SPECIALIZING IN BUSINESS AND PERSONAL COMPUTERS *

Featuring:

DATA GENERAL microNOVA®
XITAN GENERAL
APPLE II
S-100 Bus Products
SOFTWARE currently available:
AR, GL, AP, Inv., Payroll, Word Processing, and Dental Office Manager.

170 Main Street
Nashua, NH 03060
603/883-2386

microNOVA® is a registered trademark of Data General Corp

Circle 76 on inquiry card.

PET SCHEMATICS

Another First From "PET-SHACK".

For only $24.95 you get:

24" x 30" schematic of the CPU board, plus oversized schematics of the Video Monitor and Tape Recorder, plus complete Parts Layout—all Accurately and Painstakingly drawn to the Minutest detail.

PET ROM ROUTINES

Another Breakthrough From "PET-SHACK".

Complete Assembly Listings of all 7 ROMs, plus identified subroutine entry points; Video Monitor, Keyboard Routine, Tape Record and Playback Routine, Real Time Clock, etc.

To entice you we are also including our own Machine Language Monitor program for your PET using the keyboard and video display. All this for only $19.95.
SATISFACTION GUARANTEED.

You can have the Monitor program on cassette for only $9.95 extra.

Send check, money order, or Master Charge or VISA number and expiration date to:

PET-SHACK Software House
Marketing and Research Co.
P. O. Box 966
Mishawaka, IN 46544

Circle 303 on inquiry card.



Circle 213 on inquiry card.

CANADIANS

Announcing

HAMILTON LOGIC SYSTEMS

Specializing in logic devices, microprocessors, memorys, TTL, Cmos, etc.

Send for your catalogue

Box 7
STONEY CREEK
ONTARIO L8G 3X7

Circle 157 on inquiry card.

SAVE 5 to 15%

	LIST	SALE PRICE
IMSAI 8080 W/22 SLOTS	$751	$649
16K APPLE II	$1195	$1050
SHUGGART MODEL 800	$650	$595

SIMILAR DISCOUNTS ON:

- *IMSAI 4K, 16K, 32K RAM, MIO etc.*
  POLYMORPHICS VIDEO
  DYNABYTE 16K (ASM) RAM
- *HAZELTINE 1400, 1500, 1510*
  NOVATION ACOUSTIC COUPLERS
  1µS 8K (ASM) RAM
  PROCESSOR APPLICATIONS DMA FLOPPY DISC CONTROLLER (ASM)
  TARBELL FLOPPY DISC CONTROLLER
  TARBELL CASSETTE INTERFACE
  SHUGART GSI MODEL 110
  PICKLES AND TROUT MOD FOR HITACHI TV
  ALL SWTP PRODUCTS
  APPLE II
  MODEL 43 TELEPRINTER (OCTOBER)
  ADM 3A TERMINAL
  SOROC IQ120 TERMINAL
  TELETYPE MODEL 40 300 LNP
  CENTRONICS PRINTER
- *REPAIRS IN-HOUSE AT LOW COST*
- *CUSTOM SOFTWARE & FIRMWARE*

THE COMPUTER STOP

16919A Hawthorne Blvd
Lawndale CA 90260
(213) 371-4010

11:30 to 6:00 Tue-Sat
Closed Sunday/Monday

Circle 73 on inquiry card.

ACCELEWRITER DOUBLES DECWRITER SPEED

The ACCELEWRITER doubles the LA36 Decwriter's speed from 30 to 60 CPS. The ACCELEWRITER is a small molded module which plugs into the printer's logic board.

The ACCELEWRITER enables the Decwriter to print at its "catch-up' speed of 60 CPS all of the time. You must be able to program fill characters into the data stream after carriage returns.

Easy installation and easy reconversion, if ever desired. Compatible with standard Decwriter option boards. Full one-year warranty. Only $95.00.

Please write or phone.
Illinois residents add 5% sales tax.

Available NOW from

LARKS ELECTRONICS & DATA
P. O. Box 22
Skokie, Illinois 60077
(312) 677-6080

NORTH STAR PAYROLL PACKAGE

☆ Salary, hourly or commission
☆ Any number of pay periods per year
☆ Hash coding allows faster file accessing
☆ Multiple companies on one disk
☆ New tax rates are easily entered as data statements
☆ Simple to use — minimum operators response
☆ Selective printouts for employee information i.e. last pay, MTD, QTD, YTD, addresses, W-2, etc...

Unconditionally guaranteed. Can be returned for refund if not satisfied.

North Star disk — $75.00

Potter's Programs
22444 Lakeland
St. Clair Shores, MI 48081
(313) 573-8000

Circle 308 on inquiry card.

PASCAL $2995 COMPUTER

Call (714) 979-9920

MONTHLY SPECIALS

- Floppy Kit
  SA 801 floppy (8") & disk cont. kit (S-100) $599
- Add On Floppy Drives
  SA 800/1 (8") floppy $469
  SA 400 mini floppy $285
- Centronics 779 printer $999
- Miscellaneous
  CP/M Operating System/Manuals... $89
  8" Memorex Diskettes (IBM comp.) $3.50
  8" Verbatim Diskettes (IBM comp) $3.99
  4" Diskettes $3.39
  16k RAM chips $9.75
  Pwr Supply cabinet (mini floppy) ... $79
  Pwr Supply cabinet (floppy) $225

CIT COMPUTER INTERFACE TECHNOLOGY
Peripheral Products Division
2080 South Grand, Grand Centre, Santa Ana, CA 92705

Circle 74 on inquiry card.

digits from 1 to 8 seemed too difficult, I took it as an interesting problem in itself. At first I doubted that BASIC was fast enough, but as it turned out I received a permutation every second or so using Micropolis BASIC on a SOL 20. That encouraged me to complete the eight Queens problem.

My first solution appeared in less than a minute and a half. That's a lot faster than the 20 minutes Terry Smith waited! The last of the 92 solutions came out 27 minutes from the start of the run.

The line numbers in listing 1 which are multiples of 10 are the permutation program. The rest were added to screen out the diagonal attacks. The AND or OR operators operate on each bit independently, as in machine code.

Thanks for an interesting couple of hours.

Also, I am following the Pascal articles and may implement the current series, although it's too bad the data types could not be implemented. I also wish I had some idea of their speed.

Richard Greenlaw
251 Colony Ct
Gahanna OH 43230

Listing 1:

```
10 REM PROGRAM TO CALCULATE CHESS BOARD POSITIONS FOR 8 QUEENS
12 REM WITH NO ATTACKS ANYWHERE.
15 REM BY RICHARD GREENLAW 10/7/78
20 REM TESTS PERMUTATION METHOD MENTIONED BUT NOT USED BY
22 REM TERRY SMITH IN 10/78 BYTE
28 REM D VARIABLES ARE VERTICAL POSITIONS IN EACH COLUMN
30 REM F VARIABLES ARE FLAGS REPRESENTING THE DIGIT USED
32 REM U VARIABLES COMBINE FLAGS FOR ALL DIGITS USED SO FAR
100 FOR D1=1 TO 8
110   U1=2↑D1
120   FOR D2=1 TO 8
130     F2=2↑D2
140     IF (F2 AND U1)<>0 THEN 970
145     IF D2=D1+1 OR D2=D1-1 THEN 970
150     U2=F2 OR U1
160     FOR D3=1 TO 8
170       F3=2↑D3
180       IF (F3 AND U2)<>0 THEN 960
185       IF D3=D2+1 OR D3=D2-1 OR D3=D1+2 OR D3=D1-2 THEN 960
190       U3=F3 OR U2
200       FOR D4=1 TO 8
210         F4=2↑D4
220         IF (F4 AND U3)<>0 THEN 950
225         IF D4=D3+1 OR D4=D3-1 OR D4=D2+2 OR D4=D2-2 THEN 950
226         IF D4=D1+3 OR D4=D1-3 THEN 950
230         U4=F4 OR U3
240         FOR D5=1 TO 8
250           F5=2↑D5
260           IF (F5 AND U4)<>0 THEN 940
262           IF D5=D4+1 OR D5=D4-1 OR D5=D3+2 OR D5=D3-2 THEN 940
264           IF D5=D2+3 OR D5=D2-3 OR D5=D1+4 OR D5=D1-4 THEN 940
270           U5=F5 OR U4
280           FOR D6=1 TO 8
290             F6=2↑D6
300             IF (F6 AND U5)<>0 THEN 930
302             IF D6=D5+1 OR D6=D5-1 OR D6=D4+2 OR D6=D4-2 THEN 930
304             IF D6=D3+3 OR D6=D3-3 OR D6=D2+4 OR D6=D2-4 THEN 930
306             IF D6=D1+5 OR D6=D1-5 THEN 930
310             U6=F6 OR U5
320             FOR D7=1 TO 8
330               F7=2↑D7
340               IF (F7 AND U6)<>0 THEN 920
342               IF D7=D6+1 OR D7=D6-1 OR D7=D5+2 OR D7=D5-2 THEN 920
344               IF D7=D4+3 OR D7=D4-3 OR D7=D3+4 OR D7=D3-4 THEN 920
346               IF D7=D2+5 OR D7=D2-5 OR D7=D1+6 OR D7=D1-6 THEN 920
350               U7=F7 OR U6
360               FOR D8=1 TO 8
370               F8=2↑D8
380               IF (F8 AND U7)<>0 THEN 910
382               IF D8=D7+1 OR D8=D7-1 OR D8=D6+2 OR D8=D6-2 THEN 910
384               IF D8=D5+3 OR D8=D5-3 OR D8=D4+4 OR D8=D4-4 THEN 910
386               IF D8=D3+5 OR D8=D3-5 OR D8=D2+6 OR D8=D2-6 THEN 910
387               IF D8=D1+7 OR D8=D1-7 THEN 910
500 PRINT D1;D2;D3;D4;D5;D6;D7;D8
910               NEXT D8
920             NEXT D7
930           NEXT D6
940         NEXT D5
950       NEXT D4
960     NEXT D3
970   NEXT D2
980 NEXT D1
READY
RUN
 1  5  8  6  3  7  2  4
 1  6  8  3  7  4  2  5
 1  7  4  6  8  2  5  3
 1  7  5  8  2  4  6  3
 2  4  6  8  3  1  7  5
 2  5  7  1  3  8  6  4
 2  5  7  4  1  8  6  3
 2  6  1  7  4  8  3  5
 2  6  8  3  1  4  7  5
 2  7  3  6  8  5  1  4
 2  7  5  8  1  4  6  3
 2  8  6  1  3  5  7  4
 3  1  7  5  8  2  4  6
 3  5  2  8  1  7  4  6
 3  5  2  8  6  4  7  1
 3  5  7  1  4  2  8  6
 3  5  8  4  1  7  2  6
 3  6  2  5  8  1  7  4
 3  6  2  7  1  4  8  5
 3  6  2  7  5  1  8  4
```

Listing 1 continued:

```
3 6 4 1 8 5 7 2
3 6 4 2 8 5 7 1
3 6 8 1 4 7 5 2
3 6 8 1 5 7 2 4
3 6 8 2 4 1 7 5
3 7 2 8 5 1 4 6
3 7 2 8 6 4 1 5
3 8 4 7 1 6 2 5
4 1 5 8 2 7 3 6
4 1 5 8 6 3 7 2
4 2 5 8 6 1 3 7
4 2 7 3 6 8 1 5
4 2 7 3 6 8 5 1
4 2 7 5 1 8 6 3
4 2 8 5 7 1 3 6
4 2 8 6 1 3 5 7
4 6 1 5 2 8 3 7
4 6 8 2 7 1 3 5
4 6 8 3 1 7 5 2
4 7 1 8 5 2 6 3
4 7 3 8 2 5 1 6
4 7 5 2 6 1 3 8
4 7 5 3 1 6 8 2
4 8 1 3 6 2 7 5
4 8 1 5 7 2 6 3
4 8 5 3 1 7 2 6
5 1 4 6 8 2 7 3
5 1 8 4 2 7 3 6
5 1 8 6 3 7 2 4
5 2 4 6 8 3 1 7
5 2 4 7 3 8 6 1
5 2 6 1 7 4 8 3
5 2 8 1 4 7 3 6
5 3 1 6 8 2 4 7
5 3 1 7 2 8 6 4
5 3 8 4 7 1 6 2
5 7 1 3 8 6 4 2
5 7 1 4 2 8 6 3
5 7 2 4 8 1 3 6
5 7 2 6 3 1 4 8
5 7 2 6 3 1 8 4
5 7 4 1 3 8 6 2
5 8 4 1 3 6 2 7
5 8 4 1 7 2 6 3
6 1 5 2 8 3 7 4
6 2 7 1 3 5 8 4
6 2 7 1 4 8 5 3
6 3 1 7 5 8 2 4
6 3 1 8 4 2 7 5
6 3 1 8 5 2 4 7
6 3 5 7 1 4 2 8
6 3 5 8 1 4 2 7
6 3 7 2 4 8 1 5
6 3 7 2 8 5 1 4
6 3 7 4 1 8 2 5
6 4 1 5 8 2 7 3
6 4 2 8 5 7 1 3
6 4 7 1 3 5 2 8
6 4 7 1 8 2 5 3
6 8 2 4 1 7 5 3
7 1 3 8 6 4 2 5
7 2 4 1 8 5 3 6
7 2 6 3 1 4 8 5
7 3 1 6 8 5 2 4
7 3 8 2 5 1 6 4
7 4 2 5 8 1 3 6
7 4 2 8 6 1 3 5
7 5 3 1 6 8 2 4
8 2 4 1 7 5 3 6
8 2 5 3 1 7 4 6
8 3 1 6 2 5 7 4
8 4 1 3 6 2 7 5 ■
```

Call for Papers for SIGPLAN Symposium on Compiler Construction

The SIGPLAN Symposium on Compiler Construction which is scheduled for August 8 through 10 in Boulder CO has issued a call for papers. The symposium will consider compiler construction methods and construction experience. The emphasis will be less on theoretical methods and more on techniques applied to real compilers. Especially welcome are papers on the application of theoretical methods and practices to practical compilers. Some typical topics of interest are: code generation techniques; compiler generation for particular languages, language constructs, or system architectures; compiler portability and bootstrapping; compiler testing and verification; code optimization and its practical effect; measuring and achieving compile time and execution time efficiency. 6 copies of a summary (not a complete paper) should be sent to the program chairman Dr Stephen C Johnson, room 2C-559, Bell Telephone Laboratories, 600 Mountain Av, Murray Hill NJ 07974. Summaries should not indicate author name and address. This information should appear only in a cover letter to the chairman. Summaries should explain what is new and interesting about the work and what has actually been accomplished. February 1 1979 is the deadline for submission of summaries.■

National Computer Tournament for the Prisoner's Dilemma

Looking for a challenging project for your strategic instincts and your programming skills? This computer tournament may be for you. There is no charge to enter.

The tournament is based on a nifty little game called the Prisoner's Dilemma. In the Prisoner's Dilemma there are two players. Unlike most games, such as chess, the two players are not in total conflict. In fact, both can do well or both can do poorly.

Here is how the tournament works. The game will be played for an average of 200 moves, and in each move, each player can choose either to cooperate or to defect. If both cooperate, both do well. But if one defects while the other cooperates, the defecting player gets his highest payoff, and the cooperating player gets taken for a sucker and gets his lowest payoff. The catch is that if both defect, both do poorly.

The precise payoffs in the tournament for a given move are three points each if both cooperate; five points to a player who defects while the other cooperates, with zero points to the sucker; and one point each if both defect. The score of a player in a single game is his or her total over all the moves.

To win the tournament you have to get the highest total score summed over all the games you play. Therefore your object is to get a good score in each separate game, but *not necessarily* to get a better score than the player with whom you are currently playing.

To join the computer tournament you submit a program written in BASIC or FORTRAN IV which will be a decision rule for the selection of the cooperative or the defecting choice at each move. The decision rule may be based on the history of the game so far. For example, a simple and quite effective decision rule is *Tit For Tat:* cooperate on the first move, and then do exactly what the other player did on the previous move. Quite sophisticated decision rules can be written in as little as 25 lines.

This tournament is part of a research project to understand the nature of skillful performance in a 2 sided environment which is partially cooperative and partially competitive.

Each person who completes an entry will receive a report describing the results of the tournament. The winner will receive a handsome engraved trophy.

To get further details on the tournament, write to Prof Robert Axelrod, Institute of Public Policy Studies, The University of Michigan, 506 E Liberty St, Ann Arbor MI 48104.■

Using a Calculator to Perform Hexadecimal Addition and Subtraction

This helpful hint to anyone analyzing hexadecimal programming and memory dumps comes from the *NCR Century Systems Software* volume 4, issue 2, by NCR analyst Robert C Moler.

If you do not have a Texas Instruments hexadecimal calculator and work with hexadecimal numbers, you only need to remember the number 84.

Here is how it is done. Allow two decimal places for each hexadecimal character and assign the values as follows: 0=00, 1=01, 2=02, 3=03, 4=04, 5=05, 6=06, 7=07, 8=08, 9=09, A=10, B=11, C=12, D=13, E=14, F=15.

During addition, when the sum of two decimal representations of hexadecimal numbers exceeds 15 (F), add 84 to it to perform the carry to the next place.

Example:

```
Hexadecimal      Keyboard
  Number

  1B4D          01 11 04 13
 +1C85        + 01 12 08 05
  37D2          02 23 12 18
              +          84
                02 23 13 02
              +    84 00 00
                03 07 13 02 = 37D2
```

For subtraction, when a borrow is made from the next higher place the result is in the range of 85 to 99. The result is corrected by subtracting 84 from that place. Using the preceding example, subtracting 1C85 from 37D2:

```
   03 07 13 02
 - 01 12 08 05
   01 95 04 97
 -          84
   01 95 04 13
 -    84 00 00
   01 11 04 13 = 1B4D
```

Although the limitation is four digits on my 8 digit calculator, I still find this method helpful when digging through a hexadecimal dump.

Harold Pritchard
1801 Oxford St N
St Petersburg FL 33710■

Continued from page 10

tive), but the statement J:0 returns to the start of the program rather than to the last accept.

Also, two simple changes can save one level of nesting each:

```
004A 7E E1AC JIN    JMP  INEEE
004E 7E E1D1 JOUT JMP  OUTEEE.
```

All of the above are simple modifications, but I rewrote the whole program to optimize it by using RAM in the STACK area for LOC, CHR and FLG, while deleting LST.

I am sorry to propose these modifications only in MOTOROLA M6800 format but I am sure that anyone using the 8080 can work it out on their own.

Keep up the good work for 6800 experimenters.

A De Longchamp
7600 Rousselot, Apt 6
Montreal, Quebec
CANADA H2E 1Z3

A TIMELY MODIFICATION TO KIMER

We attempted to run a program from Robert Baker's article "Kimer: A KIM-1 Timer" (July 1978 BYTE, page 12). We found Mr Baker's statement that the timer would run as a 12 hour clock by changing the contents of a particular address not to be true. After incrementing time at 12:59:59, the 12 hour clock goes to 00:00:00. The proper time (01:00:00) can be displayed by changing and adding the following code:

```
0257  C9   13                CMP  #13
  .
  .
025B  20   66   02           JSR  MIDD
  .
  .
0266  A9   #01       MIDD    LDA  #01
0268  85   FB                STA  POINTH
026A  60                     RTS
```

We hope that this correction will aid others in utilizing the KIM-1 as a 12 hour digital clock for demonstrative purposes.

Alan A Leff
Donald L Boos
The Standard Oil Company
3092 Broadway Av
Cleveland OH 44115

GRAPHIC PRESENTATION

I found the articles by Jeffrey L Posdamer, "The Mathematics of Computer Graphics," and Joel C Hungerford, "Graphics Manipulations Using Matrices," quite interesting (see September 1978 BYTE, pages 22 and 156, respectively). I am pleased to see that computer graphics is becoming more popular and useful at all levels. The article by Posdamer was particularly interesting because it closely parallels the material

presented in chapter 2 of our book, *Mathematical Elements for Computer Graphics.*

There are one or two small comments that may be of interest. Posdamer states that there are three basic transformations in two dimensions: translation, rotation, and scaling. In the sense that Posdamer is using there is of course a fourth, namely, reflection.

Hungerford's statement with respect to projected views although technically correct is confusing. In real life we see in perspective, that is, all lines converge at infinity. The projected view given by Hungerford is technically an orthographic projection. In an orthographic projection all parallel lines remain parallel. Orthographic projections are commonly used for engineering drawings. It is this type of projection that he describes.

David F Rogers PhD
Prof of Aerospace Engineering
Director Computer Aided Design
Interactive Graphics
US Naval Academy
Annapolis MD 21402

TINY PASCAL

I would like to take this opportunity to thank you for publishing the series of articles about "A 'Tiny' Pascal Compiler" (September, October and November 1978 BYTE) I have enjoyed them very much. I have considered (and am still considering) obtaining UCSD Pascal, but I do not own a disk system running CP/M. This is why I have appreciated your series of articles, since they did not require a disk system and could be easily modified for any 8080 system (I own a MECA cassette operating system). I hope you will continue having serious articles about Pascal: by this I mean articles perhaps extending Tiny Pascal's features compared with full standard Pascal and about structured programming in Pascal or Pascal-like languages; I think I have seen enough articles in all the personal computing magazines lately comparing the virtues and vices of Pascal vs BASIC and the like.

Harvey E Hahn
14 N Highview
Addison IL 60101

CHESS 4.6 DEFEATED?

In reference to Norman Whaland's article "A Computer Chess Tutorial" (October 1978 BYTE, page 168): Chess 4.6 proved its weakness when it lost to Duchess at the Jerusalem Conference on Information Technology Computer Chess Tournament on August 6 thru 9 1978. Chess 4.6 resigned after 59 moves (copies of that game are available upon request along with a self-addressed stamped envelope). Duchess had a clear lead well into midgame, but played rather aimlessly, picking off Chess 4.6's pieces one at a time and leaving a lone King to resign. The final standings were as follows:

1. Duchess	3 points
2. Chess 4.6	2 points
3. Chaos	2 points
4. Ostrich	1 point
5. Tell	1 point
6. BS '66 '76	0 points

In addition, I have a program written in Pascal to solve the 8 Queens problem. It was run on the Hebrew University computer (CDC 64000) with the original Pascal compiler: ETH Zurich (Release 2). The program works recursively and contains more comments than actual coding to make understanding easy. It too is available upon request (with 30 cents in stamps to cover shipping). Also available are a Magic Squares generator and a Life generator written in Pascal.

Henry Nussbacher
570 Fort Washington Av
New York NY 10033

HEATH BIT

I recently purchased a DECwriter (H36) for my H8 system which already included an H9 terminal. Difficulties ensued when I tried to list lower case text (such as this letter) on the H9, since it interprets ASCII code 140 thru 177 octal as 040 thru 077 octal. A remedy for this situation was published in September 1978 BYTE, page 147, by George J Frye, using an inverted bit 7 in place of bit 6 from the ASCII bus to feed the H9 character generator. Unfortunately, the method described would cause an erroneous signal to be sent to the plot mode decoder, thus making that function of the H9 useless.

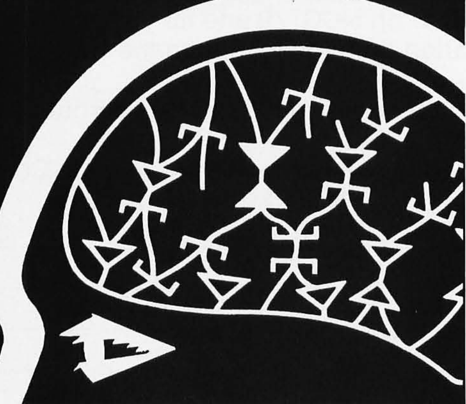
Happily the fix suggested by Mr Frye need only be slightly modified in order to preserve the PLOT function and allow lower case ASCII to be displayed as upper case:

1. Cut the foil trace from pin 1 of IC206 to pin 22 of IC205.
2. Jumper from pin 10 of IC203 to pin 11 of IC219.
3. Jumper from pin 10 of IC219 to pin 22 of IC205.

I have made this change to my H9 and it works flawlessly. After I had completed it, however, I noticed that the inverted ASCII bit 7 signal was already available from IC203. Thus, the unused segment of the hexadecimal inverted (IC219 pins 10 and 11) need not be used, and only one jumper is needed:

1. Cut the foil trace from pin 1 of IC206 to pin 22 of IC205.
2. Jumper from pin 11 of IC203 to pin 22 of IC205.

William W Moss
1507 Riverview Ln
Bradenton FL 33505



CLARIFYING THE LIMIT

There are a number of items in "The Sky's the Limit" (November 1978 BYTE, page 48) which require clarification.

On page 50 it is stated that "some repeaters have reverse call-in capability by which the nonamateur can dial the repeater." The first part of the sentence is true. The second part is incorrect. FCC regulations do not allow a nonlicensed person to activate an amateur transmitter. It should be realized that the repeater is an amateur transmitter and is licensed.

It should be further noted that it is not legal for an amateur to directly transmit messages to a nonlicensed person; most amateurs would be reticent to do so. Third party telephone communication is a special exception. If one would monitor an amateur VHF repeater, he would find that the amateur in his car wishing to use "autopatch" first accesses the repeater (the licensed station) which is under direct supervision of control operators, also licensed. Only then does the amateur place his desired *nontoll* telephone call using Touch Tone signaling.

In response to phrases like "As radio amateurs recognize the capabilities of the microcomputer. . ." (page 50), I can only state that I know a number of radio amateurs who have been working with microcomputers before there were "home computers." I really don't feel that the radio amateur belongs in the class of the slow learner. However, I agree with the statement on page 61 that "the AMSAT OSCAR amateur radio communication satellites can provide the computer enthusiast with an operationally simple worldwide personal computing network"—but if and only if that enthusiast is also a licensed radio amateur.

To Mr Kasser I would like to add:

1. A review of FCC rules and regulations may be appropriate.
2. If you are really interested in a multidiscipline approach (ham radio and computers)—take the time to get a license and enjoy both worlds, as many of us do.

G William Pfeiffer
K0GVX
2122 Novak Av N
Stillwater MN 55082

*A reference is made on page 50 of the article to the activation of an amateur VHF repeater by a nonamateur computer user. This procedure is commonly known as "reverse autopatch." The Federal Communications Commission has ruled that such activation may be done only in the following way: the nonamateur dials a telephone number which rings a phone at the repeater control site. A human licensed amateur radio operator answers the call, establishes its purpose, and determines if transmission of the call is consistent with the regulations governing amateur radio operation. The repeater operator may then turn on the transmitter and allow transmission of the nonamateur's message under supervision of the amateur. The repeater operator must also keep a record of the transmission to comply with the rules of third party traffic. . . .*RS, N4ANG

HOUSE OPERATING SYSTEM

I read the editorial in the October 1978 BYTE and noted with interest the distribution of interests in your survey. Naturally, I was cheered to see home automation ranking as number two in the list, since it is precisely what I am working on.

The "House Operating System" is really a complete, general purpose power control system that can be installed in almost any reasonable environment for any desired purpose. It can deal with both on and off controls such as ordinary lights, and controls that require some parametric input, such as a digitally controlled FM tuner.

What most concerns me, however, is the human engineering of the system. It has full programming capabilities in most respects, interacts with the environment and user, and guides the user in its use.

The basic input device in the House Operating System is either a 16 button telephone—like keypad or a small alphanumeric keypad (about the size of a calculator) connected to the central computer by a pair of twisted wires. Any number of these units can be connected on a single pair. All input is echoed on video connected thru a master antenna system, enabling all televisions in the house to be output terminals. Prompting messages, etc, are also logged on this video output.

Commands in the House Operating System are mnemonic, and defined by the user. They are set up when the system is installed, and can be changed, deleted, or added at any time.

Many primitive commands come with the basic software: define a command to turn a single relay on or off, program a command string, define parameters, conditions, etc.

The system is available only as a turnkey package, but we will sell the power control boards separately. Our optoisolator board (for input), tone receiver board (for 16 button keypad decoding), and relay driver boards (power control) are bus independent, communicating through serial and parallel ports, and are thus usable on any system.

Frank Alviani
GIMIX Inc
1337 W 37th Pl
Chicago IL 60609■

Circle 132 on inquiry card.

DeskTop Wonders

Some Random Games

C K Adams
12702 Shady Creek
St Louis MO 63141

With the increasing number of inexpensive programmable calculators, game and entertainment programs are becoming popular.

Here are two simple entertainment programs written for the Texas Instruments model 58 or 59 programmable calculators with the master library. After entering any of the programs, always single step through the program, checking the entries against the program listing to help prevent wrong entries.

Listing 1 is the ESP program. The object of this game is to guess the number (0 or 1) that the program will come up with. The operator enters either a 0 or a 1 into the calculator, then presses D. The calculator displays its number, independent of the input. The number of correct guesses is stored in register 01 while the total number of guesses is stored in register 00.

The random number generator program of the master library is used to calculate a random number between 0 and 1. The program then rounds the number to either a 0 or 1. The steps required to generate this random number using the library program are as follows:

1. Call up program 15.
2. Enter a seed number.
3. Press A.
4. Press E to initialize the program.
5. Press SBR and D.MS to obtain answer.

To obtain another random number, repeat step 5. These steps are accomplished by program steps 00, 01, 03 through 05, and 12 through 15.

The guess is entered by steps 08 through 11 and is stored in register T. Steps 21 and 22 keep track of the number of guesses. Steps 23 through 25 determine if the guess is correct, and steps 26 and 27 keep track of the number of correct guesses.

To use the program:

1. Enter the program from the program listing.

2. Press RST.
3. Enter a seed number (between 0 and 19907).
4. Press A.
5. Enter guess (either 0 or 1).
6. Press D (the number calculated will then be displayed).
7. Repeat steps 5 and 6 for the next guess.
8. Read registers 00 and 01 to determine the number of guesses and the number of correct guesses.
9. Start at step 2 to reinitialize the program.

Listing 2 is a dice game. This program allows the user to "roll two dice" and end up with two random numbers (representing the dice) between 1 and 6. The two numbers are separated by a decimal point. In this program, as with the previous program, a seed number is entered to initialize the random number generator. Entering the same seed number will result in the same sequence of numbers for the rolls of the dice. But there are over 19,000 possible combinations for the seed, so it is virtually impossible to memorize the sequence for more than a few seed numbers. Also, by letting a player other than the operator enter the seed number, the possibility of cheating is eliminated.

Program steps 1 through 14 set the limits for the random number program to 0.5 and 6.5. Steps 16 and 17 set up the loop for the roll of the dice. Steps 21 through 23 determine the value of the first die. Steps 24 through 30 round this to a whole number

Listing 1: ESP program for guessing which number will be played next.

Location	Keys	Commentary
00	2nd LBL A	Define A as seed.
02	2nd CM	Clear memory.
03	2nd PGM 15 E	Initialize random number program.
06	2nd LBL +	Define loop.
08	R/S 2nd LBL D	Enter guess.
11	X⇆T	Move to T.
12	2nd PGM 15 SBR 2nd D.MS	Calculate random number between 0 and 1.
16	2nd FIX 0 EE INV EE	Round answer to either 0 or 1.
21	2nd OP 20	Increment number of guesses.
23	INV 2nd X=T +	If guess does not equal number, go to loop.
26	2nd OP 21	Increment number of correct guesses.
28	GTO +	Go to loop for next try.

and store the results. Steps 31 through 38 determine the value of the second die and round it to a whole number. Steps 39 through 46 combine the two numbers by multiplying the value of the second die by 0.1 and adding to the first value. Steps 47 and 48 loop back to display the results and wait for the next roll.

To use the program:

1. Load program from program listing.
2. Press RST.
3. Enter seed number (between 0 and 19907).
4. Press R/S (display will go out and then display 0).
5. Press D to roll the dice. The display will show the value of the two dice, separated by a period.
6. Press D to roll the dice again.
7. To reinitialize the program, return to step 2.■

Listing 2: Dice program for rolling two dice. The two dice are displayed on either side of the decimal point.

Location	Keys	Commentary
00	2nd CM	Enter seed number.
01	2nd PGM 15	Initialize random number program.
03	E . 5	Set lower limit at 0.5.
06	2nd PGM 15	
08	A 6 . 5	Set upper limit at 6.5.
12	2nd PGM 15 B 0	Zero display.
16	2nd LBL +	Set up loop.
18	R/S	Stop program and display results.
19	2nd LBL D	
21	2nd PGM 15 C	Calculate value of first die.
24	2nd FIX 0 EE INV EE	Round value to a whole number.
29	STO 0	Store value of first die.
31	2nd PGM 15 C	Calculate value of second die.
34	2nd FIX 0 EE INV EE	Round value to a whole number.
39	INV 2nd FIX X . 1	Multiply value of second die by 0.1.
44	+ RCL 0 =	Combine first and second dice values.
47	GTO +	Loop back for next roll.

GOTOlocks and the Three Sorts

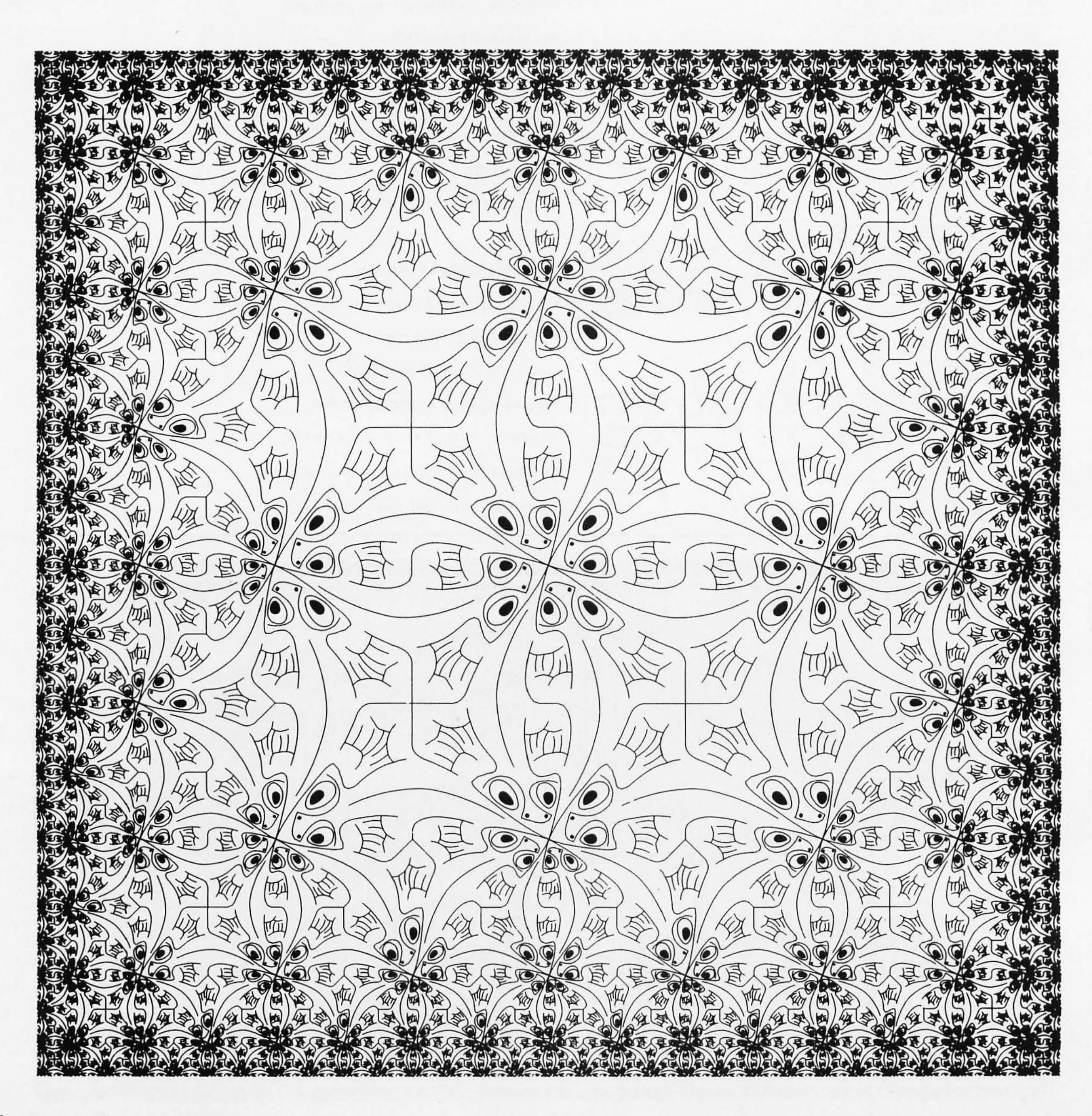
Gwen Hadley
520 E May, Apt 2
Las Cruces NM 88001

In a package program on the dump of a null allocation, there resided a very asynchronous VERB who was commonly known, in reverse Polish notation, as GOTOlocks. GOTOlocks was a virtual card hopper, a veritable TALLY-goer, a member of the MOD function, for she was continually GOTOing. Her run time was so elliptic that there was no algorithm for tracing her or her Turing machine.

"I am queried of this dump routine," she asserted. "To GOTO is Greater Than to RETURN." And her counter was dimensioned around that password.

One day GOTOlocks multivibrated the topdown on her Turing machine, implemented her ascending key, converted her scale factor for sequential access, made a scientific notation of her default value, set her level indicator at zero, and linked out into the recursive overflow area of the installation. "I will have a real time for myself," she stated.

Soon, her Turing machine arrived at an



empty operating system which belonged to the Three Sorts. There was a Major Sort, an Intermediate Sort and a Minor Sort. The Three Sorts were just taking a structured walk-through the forest while their hash was being debugged.

GOTOlocks inputted into the Three Sorts' priority system through their OR gate and advanced to their hash table. She sampled the Major Sort's hash.

"There are too many bugs in it," she asserted. "An adder would take care of them." She correlated four bits of cyberculture into the floating point. "At least it can be upgraded by integrating a literal queuing solution," she commented.

NEXT, GOTOlocks tested the Intermediate Sort's hash. "This is too undefined," she printed. "No wonder it isn't labeled. It could be a key punch with a little binary addition." She loaded some Garbage In to it.

Then, GOTOlocks sampled the Minor Sort's hash. "What a smooth approximation!" she declared. And she simulated the entire batch.

NEXT, GOTOlocks multivibrated into the Three Sorts' mass storage. Here, she found three peripheral devices—a big peripheral device, a medium peripheral device and a tiny peripheral device.

She tried the big peripheral device. "This is too ambiguous," she stated, "I am Not Equal to it. All I need is a simple basic cycle."

She tested the medium peripheral device. "This is too digital," she commented, "and with its variable flip flopping I feel as if I am loaded. There is no use making myself Illiac."

She tried the tiny peripheral device, and it was just right. Just as she began to operate it, however, it was truncated.

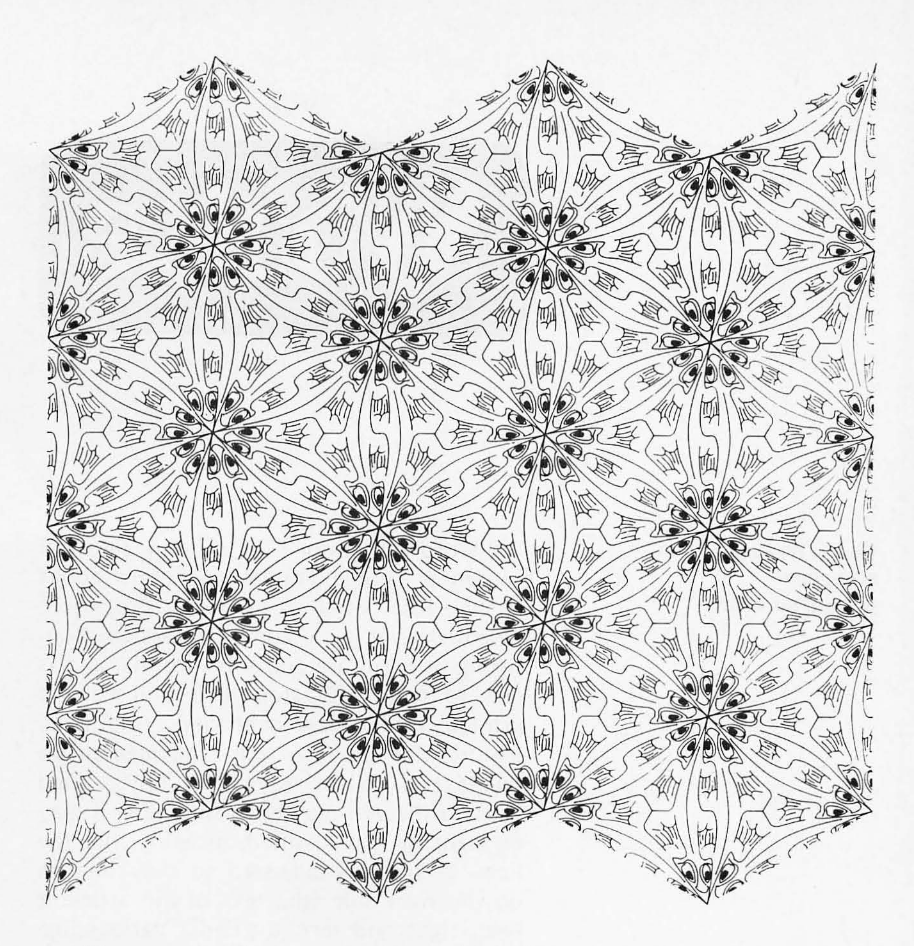
Finally, GOTOlocks entered the Three Sorts' deck setup. She tried the huge source deck, but it was too dimensional. "What I need," she declared, "is something more finite."

She tried the medium source deck, but it had an embedded parenthesis in it. "This could give one a BACAIC," she stated, and moved on to the tiny source deck.

The tiny source deck was so very heuristic that GOTOlocks was soon smoothing away.

Suddenly, the Three Sorts returned from their walk-through.

"Someone has been processing my hash," asserted the Major Sort.



"Someone has been processing my hash," declared the Intermediate Sort.

"Someone has been processing my hash," iterated the Minor Sort, "and it is all simulated."

They went into their mass storage.

"Someone has operated my peripheral device," asserted the Major Sort.

"Someone has operated my peripheral device," declared the Intermediate Sort.

"Someone has operated my peripheral device," iterated the Minor Sort, "and it has been truncated."

Finally, they linked into their deck setup. "Someone has been scaling in my source deck," asserted the Major Sort.

"Someone has been scaling in my source deck," declared the Intermediate Sort.

"Someone has been scaling in my source deck," iterated the Minor Sort, "and here she is!"

With that, GOTOlocks became on line and listed through the source deck, and out through the mass storage, and through their priority system with the Sorts vibrating on her trail.

She looped off in her Turing machine, taking direct access to her dump routine. "One would conclude that I am linked to the 5th column," she stated, making a scientific notation never to go into a recursive overflow area again.■

Art by Lynn Irwin
717 Emery St
Longmont CO 80501

Here is a little contest, suggested by the level of punmanship exemplified by this short story. To enter, simply identify each buzzword buried in this story, write a short characteristic explanation of its use and normal context, and return it to us at BYTE magazine, 70 Main St, Peterborough NH 03458, attn: GOTOlocks. But beware, some of the buzzwords we haven't even heard of.

Table 1 (opposite): Commands supported by the Micro Word Processor along with the options available.



A Micro Word Processor

Theron Wierenga
POB 2007
Holland MI 49423

"Word processor" is a term personal computer people are hearing more about lately. Just what is a word processor? Essentially, it is a computer system consisting of hardware and software that allows one to enter and store text and have it printed back upon command. Additional features let the user edit and manipulate the text in various ways. The copy printed back to the user may be as entered or it may be right margin justified, meaning that the lines of text are adjusted so they line up on the right side (the text of this article is both right and left justified). Various line widths are usually possible. Page numbers and headings may be automatically added to the text, and indices and tables of contents may be generated using key words.

If you type a 20 page report, you will probably make some errors. The report may have to be retyped several times before the final copy is acceptable. With a word processing system, only the errors and changes would have to be retyped into the system — a great timesaver if the text is lengthy.

The typical microcomputer hardware for such a system might consist of a mainframe computer, 32 K bytes of memory, dual floppy disks, and a printer with proportional spacing and upper and lowercase type. A video terminal is also a useful addition for entering and editing text. The cost of such a system can run from $7000 to $12,000. Many businesses are finding such word processing systems to be valuable assets to their staffs, and well worth the investment. This is especially true if much of the hardware is already available and being used on such tasks as payroll and accounting. But what about the person who would just like to produce a few form letters or multiple copies of a computer club newsletter? And what about those of us who don't type well? Wouldn't it be nice to have a word processor that allows us to correct the

SUMMARY OF COMMANDS

Command	Options Available
C: clear file area.	
E: enter the editor.	2,3,5,6.
I: input lines.	3,4,5,6.
J: jump to top of monitor.	
L: list file as entered with line numbers.	1,3.
M: return end of file address in memory.	
P: print file as entered or justified.	1,3,6.

Options

1. CTRL C: abort printout.
2. CTRL I: tab to next selected character.
3. CTRL S: halt printout until a second CTRL S is sent.
4. CTRL X: erase current line, reissue > prompt.
5. DEL : delete last character typed in, echos underline or back arrow.
6. \ : a backslash decodes as a carriage return and line feed in the print justified routine.

Listing 1: Typical session with the Micro Word Processor system. Comments regarding the listing are in italics.

(The first lines are input into the file. Note use of DEL to erase errors in typing.)

```
COMMAND? I
>10     We hold these truths to be self-evident: that all men are created equal,
>12 that they are endowed by their c_Creator with certain un_inalienable rights,
>14 that among these are life, and the persuit of happiness.   Thatto__ to
>16 secure these rights _, governments are instituted among men, deriving their
>18 just powers from the consent of the governed.
>
```

(The file is listed back to the user.)

```
COMMAND? L

10      We hold these truths to be self-evident: that all men are created equal,
12 that they are endowed by their Creator with certain inalienable rights,
14 that among these are life, and the persuit of happiness.   That to
16 secure these rights, governments are instituted among men, deriving their
18 just powers from the consent of the governed.
```

(The word liberty is missing from line 14. The edit routine is used to insert it.)

```
COMMAND? E
LINE NO.? 14
CHAR.? ,
that among these are life, liberty, and the persuit of happiness.   That to
>
```

(The file is then printed back, using a 50 space maximum line width.)

```
COMMAND? P
COPIES(01-99)-01
JUSTIFY? Y
WIDTH(01-80)-50

     We hold these truths to be self-evident: that
all men are created equal, that they are endowed
by their Creator with certain inalienable rights,
that among these are life, liberty, and the
persuit of happiness.   That to secure these
rights, governments are instituted among men,
deriving their just powers from the consent of the
governed.
```

(The file is printed again using an 80 space line. Two copies are requested.)

```
COMMAND? P
COPIES(01-99)-02
JUSTIFY? Y
WIDTH(01-80)-80

     We hold these truths to be self-evident: that all men are created equal,
that they are endowed by their Creator with certain inalienable rights, that
among these are life, liberty, and the persuit of happiness.   That to secure
these rights, governments are instituted among men, deriving their just powers
from the consent of the governed.

     We hold these truths to be self-evident: that all men are created equal,
that they are endowed by their Creator with certain inalienable rights, that
among these are life, liberty, and the persuit of happiness.   That to secure
these rights, governments are instituted among men, deriving their just powers
from the consent of the governed.
```

(A title is entered.)

```
COMMAND? I
>08             THE DECLARATION OF INDEPENDENCE\\\
>
```

Listing 1 continued:

```
COMMAND? P
COPIES(01-99)-01
JUSTIFY? N

                 THE DECLARATION OF INDEPENDENCE\\\
     We hold these truths to be self-evident: that all men are created equal,
that they are endowed by their Creator with certain inalienable rights,
that among these are life, liberty, and the persuit of happiness.   That to
secure these rights, governments are instituted among men, deriving their
just powers from the consent of the governed.
```

(The file is printed back as entered, without line numbers.)

```
COMMAND? P
COPIES(01-99)-01
JUSTIFY? Y
WIDTH(01-80)-80

                 THE DECLARATION OF INDEPENDENCE


     We hold these truths to be self-evident: that all men are created equal,
that they are endowed by their Creator with certain inalienable rights, that
among these are life, liberty, and the persuit of happiness.   That to secure
these rights, governments are instituted among men, deriving their just powers
from the consent of the governed.
```

(The file is printed back justified, with an 80 space line width.)

```
COMMAND? M
21ED

COMMAND? J
?D
2050
21EF
2050 FF FF FF FF FF 34 08 20
2058 20 20 20 20 20 20 20 20
2060 20 20 20 20 20 20 54 48
2068 45 20 44 45 43 4C 41 52
2070 41 54 49 4F 4E 20 4F 46
2078 20 49 4E 44 45 50 45 4E
2080 44 45 4E 43 45 5C 5C 5C
2088 0D 50 10 20 20 20 20 20
2090 57 65 20 68 6F 6C 64 20
2098 74 68 65 73 65 20 74 72
20A0 75 74 68 73 20 74 6F 20
20A8 62 65 20 73 65 6C 66 2D
20B0 65 76 69 64 65 6E 74 3A
20B8 20 74 68 61 74 20 61 6C
20C0 6C 20 6D 65 6E 20 61 72
20C8 65 20 63 72 65 61 74 65
20D0 64 20 65 71 75 61 6C 2C
20D8 0D 4A 12 74 68 61 74 20
20E0 74 68 65 79 20 61 72 65
20E8 20 65 6E 64 6F 77 65 64
20F0 20 62 79 20 74 68 65 69
20F8 72 20 43 72 65 61 74 6F
2100 72 20 77 69 74 68 20 63
2108 65 72 74 61 69 6E 20 69
2110 6E 61 6C 69 65 6E 61 62
2118 6C 65 20 72 69 67 68 74
2120 73 2C 0D 4E 14 74 68 61
2128 74 20 61 6D 6F 6E 67 20
2130 74 68 65 73 65 20 61 72
2138 65 20 6C 69 66 65 2C 20
2140 6C 69 62 65 72 74 79 2C
2148 20 61 6E 64 20 74 68 65
2150 20 70 65 72 73 75 69 74
2158 20 6F 66 20 68 61 70 70
2160 69 6E 65 73 73 2E 20 20
2168 20 54 68 61 74 20 74 6F
2170 0D 4C 16 73 65 63 75 72
2178 65 20 74 68 65 73 65 20
2180 72 69 67 68 74 73 2C 20
2188 67 6F 76 65 72 6E 6D 65
2190 6E 74 73 20 61 72 65 20
2198 69 6E 73 74 69 74 75 74
21A0 65 64 20 61 6D 6F 6E 67
21A8 20 6D 65 6E 2C 20 64 65
21B0 72 69 76 69 6E 67 20 74
21B8 68 65 69 72 0D 30 18 6A
21C0 75 73 74 20 70 6F 77 65
21C8 72 73 20 66 72 6F 6D 20
21D0 74 68 65 20 63 6F 6E 73
21D8 65 6E 74 20 6F 66 20 74
21E0 68 65 20 67 6F 76 65 72
21E8 6E 65 64 2E 0D FF 40 00
?
```

(The memory function shows the address of the end of the file. The jump function returns control to the system monitor where a memory dump of the file is done. The file begins at hexadecimal 2055 and ends at hexadecimal 21ED.)

mistakes as we type? When finished, our computer could type back the page to us without any errors.

The desire for such a system led me to develop my own word processor. It is designed to run on a minimal system utilizing the 8080A type microprocessor. The computer will need only a single serial input/output (IO) port, some sort of hardcopy printer, and about 4 K bytes of memory, although less could be used.

The Micro Word Processor is a machine language program that occupies 1 K bytes of memory space. It was designed to fit into a 2708 type programmable read only memory. The text typed in is stored sequentially in programmable memory. The program is line oriented to make editing easier. Various line widths can be specified when the text is printed back to the user.

The Micro Word Processor recognizes seven different commands. They are: C (clear), J (jump), I (input), E (edit), L (list), M (memory), and P (print). A summary of the commands and available options is given in table 1.

The *clear* command clears the file area making room for new text to be entered. The *jump* command transfers program control to the system monitor located elsewhere in memory. The *input* command allows lines with line numbers to be entered into the file. Regardless of the order in which lines are entered, the program will place the lines in ascending line number order. The *edit* routine is used to correct errors or make changes in individual lines. The line number and a key character are supplied and edit slews through a line until the character is found. Additional characters can then be entered into the line or deletions made. *List* dumps out the entire file with line numbers as it was entered into the system. The *memory* function returns the address of the last character in the file. The *print* routine can either type back the file as entered without the line numbers, or print it with the user specifying the maximum allowable line width. The Micro Word Processor does not right justify, but it does check to see if the next word will fit within the allowable margins. If not, it sends a carriage return and line feed to the terminal and then continues printing. A typical session using the Micro Word Processor is shown in listing 1.

The 1 K byte Micro Word Processor, written for the 8080/Z-80 and capable of being implemented in read only memory, plus further documentation, is available for $4 postage paid from BYTE. Please use the coupon below and order BYTE document number 102.■

Please send _____ copies of BYTE Nybble #______ at $________ postpaid.

_____ *Check Enclosed*   *Signature* ____________________

Bill my BAC # ____________________ *Exp Date* __________

Bill my MC # ____________________ *Exp Date* __________

Name ______________________________

Street ______________________________

City ________________ **State** ________ **Zip Code** ________



BYTE Nybbles Library, 70 Main St, Peterborough NH 03458

You may photocopy this page if you wish to keep your BYTE intact.

Programming Quickies

Single Stepping the 8080 Processor

Charles Sharp
POB 3879
Carmel CA 93921

This is a description of a routine for the 8080 processor to allow single stepping with software (see listing 1).

Often it is useful to be able to step through a program one step at a time. To do this from a software standpoint, it is necessary to have a register save area. In this way, the registers can be stored, and the processor registers are freed to allow external input/output (IO) processing.

One straightforward approach is to insert a restart byte at each point, thus generating a branch to the register save routine and back into the monitor. However, this precludes stepping through read only memory, and the user must keep in mind where the next restart is to be placed.

An alternate approach is to have a programmable memory execution area where the program is transferred, byte by byte, and then executed under controlled conditions. The program needs to know the number of bytes (one, two or three) of the op code being simulated. In the accompanying program (see listing 1), a table lookup is used to extract this information.

The code in listing 1 does not have any provision to display the register save area. Therefore, this code must be used as part of a more complete monitor system.

The basic idea is to initialize EXOP1 and EXOP2 to NOPs. A jump to RSTA is put in EXOP3. The one, two or three bytes of code is moved to EXOP EXOP1, and EXOP2. The registers are restored to their saved values and a jump is made to EXOP. Then all the registers are immediately saved by the jump to RSTA.

Then things become more interesting. What does one do about the jump, call, return and restart statements? After all, they do have some usefulness. Well, it is obvious that we cannot actually let the jump instruction go on to its original loca-

Listing 1: A program for single stepping the 8080 processor.

```
            ORG    1FD0
1FD0 EXOP   DS     1                      RAM EXECUTION AREA
1FD1 EXOP1  DS     1
1FD2 EXOP2  DS     1
1FD3 EXOP3  DS     1
1FD4 EXOP4  DS     2
1FD6 MSTC2  DS     2                      ADDRESS AND STACK SAVE LOCATION
1FD8 FSAVE  DS     1                      FLAG SAVE LOCATION
1FD9 ASAVE  DS     1                      A REGISTER SAVE LOCATION
1FDA CSAVE  DS     1                      C REGISTER SAVE LOCATION
1FDB BSAVE  DS     1                      B REGISTER SAVE LOCATION
1FDC ESAVE  DS     1                      F REGISTER SAVE LOCATION
1FDD DSAVE  DS     1                      D REGISTER SAVE LOCATION
1FDE LSAVE  DS     1                      L REGISTER SAVE LOCATION
1FDF HSAVE  DS     1                      H REGISTER SAVE LOCATION
1FE0 SSAVE  DS     2                      STACK POINTER SAVE LOCATION
1FE2 PSAVE  DS     2                      PROGRAM COUNTER SAVE LOCATION
     MSTCK  EQU    FSAVE
            ORG    1000
1000        LXI    SP     MSTCK    31D81F SET UP STACK POINTER
1003        JMP    SSTEP           C38310 ENTRY TO SINGLE STEP
*******************************************************************
*     TABLES USED TO LOOK UP CERTAIN OPCODES AND JUMP LOCATIONS   *
*******************************************************************
1006 OOTBL  DB                     E9     PCHL INSTRUCTION BYTE
1007        DW     PCH             2B11
1009        DB                     C3     JMP INSTRUCTION BYTE
100A        DW     BRTYP           3411
100C        DB                     CD     CALL INSTRUCTION BYTE
100D        DW     CALTY           4711
100F        DB                     C9     RET INSTRUCTION BYTE
1010        DW     RETYP           5511
1012        DB                     76     HLT INSTRUCTION BYTE
1013        DW     FINI            7D10
1015 SYTBL  DW     RETYP           5511
1017        DW     EXCYC           CB10
1019        DW     BRTYP           3411
101B        DW     EXCYC           CB10
101D        DW     CALTY           4711
101F        DW     EXCYC           CB10
1021        DW     EXCYC           CB10
1023        DW     RSTTY           6C11
1025 CLTBL  DB     75591559355915593D591D593D591D59
1045 CHTBL  DB     5FD95CF95ED94EC95DD95DC95DD95DC9
*******************************************************************
*        RETURN POINT AFTER SIMULATION OF OPCODE IN RAM AREA      *
*******************************************************************
1065 RSTA   SHLD   LSAVE           22DE1F SAVE H
1068        STA    ASAVE           32D91F TEMP SAVE A
106B        RAR                    1F     ROTATE CY TO ACCUMULATOR
106C        LXI    H      0000     210000 GET 0'S IN H-L
106F        DAD    SP              39     GET VALUE OF STACK POINTER
1070        SHLD   SSAVE           22DF1F SAVE VALUE OF STACK POINTER
1073        RAL                    17     RESET CY TO PROPER VALUE
1074        LDA    ASAVE           3AD91F GET VALUE OF A
1077        LXI    SP     LSAVE    31DE1F SET STACK POINTER TO LSAVE
107A        PUSH   D               D5     SAVE D-E
107B        PUSH   B               C5     SAVE B-C
107C        PUSH   PSW             F5     SAVE PSW
107D FINI   LXI    SP     MSTCK    31E81F RESET STACK POINTER
1080        JMP    DSPLY           C3???? USER DEFINED DISPLAY ROUTINE
                                          AND LOOP CONTROL
*******************************************************************
*                    SINGLE STEP LOOP CYCLE                       *
*******************************************************************
1083 SSTEP  MVI    A      C3       3EC3   GET JUMP OPCODE
1085        STA    EXOP3           32D31F STORE IN RAM EXECUTION AREA
1088        LXI    H      0000     210000 GET 0'S IN H-L
108B        SHLD   EXOP1           22D11F MAKE 2ND & 3RD BYTES NOP'S
108E        LXI    H      RSTA     216510 GET RETURN POINT IN H-L
1091        SHLD   EXOP4           22D41F STORE IN RAM EXECUTION AREA
1094        LHLD   PSAVE           2AF11F LOCATION OPCODE TO BE SIMULATED
1097        MOV    A      M        7E     GET THE OPCODE BYTE
1098        STA    EXOP            32D01F STORE IT IN RAM EXUCUTION AREA
109B        CALL   TCNT            CDD810 GET NUMBER OF BYTES FOR OPCODE
109E        LHLD   PSAVE           2AF11F GET PROGRAM COUNTER IN H-L AGAIN
10A1        CPI    01              FE01   SEE IF ONE BYTE INSTRUCTION
10A3        JZ     NMB             CAB510 IF SO, UPDATE PSAVE
10A6        CPI    02              FE02   SET Z FLAG - TWO BYTE INSTRUCTION
10A8        INX    H               23     POINT TO SECOND BYTE
10A9        MOV    A      M        7E     FETCH THE SECOND BYTE
10AA        STA    EXOP1           32D11F STORE IN RAM EXECUTION AREA
10AD        JZ     NMB             CAB510 Z SET IF TWO BYTE INSTRUCTION
10B0        INX    H               23     POINT TO THIRD BYTE
10B1        MOV    A      M        7E     FETCH THE THIRD BYTE
10B2        STA    EXOP2           32D21F STORE IN RAM EXECUTION AREA
10B5 NMB    INX    H               23     INCREMENT TO PROPER VALUE
10B6        SHLD   PSAVE           22F11F STORE NEW PROGRAM COUNTER
10B9        LDA    EXOP            3AD01F GET THE OPCODE BYTE AGAIN
10BC        LXI    H      COTBL    210610 SET H-L TO BEGINING OF OOTBL
10BF        MVI    B      05       0605   SET COUNT OF NUMBER OF ENTRIES
10C1        CALL   TBCHK           CD0711 SEE IF IN TABLE
10C4        ANI    C0              E6C0   MUST NOT BE, GET HIGH OCTAL PART
10C6        CPI    C0              FEC0   SEE IF IT IS THREE
10C8        JZ     SYSOP           CA1911 JUMP TO SYSTEM TYPE OPCODES IF SO
*******************************************************************
*        NORMAL REGISTER AND STACK POINTER RESTORING SEQUENCE     *
*******************************************************************
10CB EXCYC  POP    PSW             F1     RESTORE PSW
10CC        POP    B               C1     RESTORE B-C
10CD        POP    D               D1     RESTORE D-E
10CE        LHLD   SSAVE           2ADF1F GET VALUE OF STACK POINTER
10D1        SPHL                   F9     RESTORE STACK POINTER VALUE
10D2        LHLD   LSAVE           2ADE1F RESTORE H-L
10D5        JMP    EXOP            C3D01F EXECUTE THE OPCODE
*******************************************************************
*                 RETURN NUMBER OF BYTES FOR OPCODE               *
*         A - HAS OPCODE ON ENTRY AND BYTE COUNT ON EXIT          *
*******************************************************************
10D8 TCNT   LXI    H      CLTBL    212510 GET 1ST PART OF OP COUNT TABLE
10DB        CPI    40              FE40   SEE IF LOW PART OF OP COUNT TABLE
10DD        JC     MAKCN           DAEB10 IF SO NEED TO LOOK UP THE COUNT
10E0        LXI    H      CHTBL    214510 GET SECOND PART OF OPCOUNT TABLE
```

Listing 1, continued:

```
10E3       SUI   C0            D6C0   SEE IF 2ND PART OF OP COUNT TABLE
10E5       JNC   MAKCN         D2EB10 IF SO NEED TO LOOK UP THE COUNT
10E8       MVI   A     01      3E01   IF NOT, COUNT MUST BE 1
10EA       RET                 C9     AND NO MORE TO DO
10EB MAKCN CPI   04            FE04   SEE IF LESS THAN OR EQUAL TO 3
10ED       JC    EXBYT         DAF610 IF SO EXTRACT THE BYTE
10F0       INX   H             23     POINT H TO NEXT BYTE
10F1       SUI   04            D604   SUBTRACT 4 FROM ACCUMULATOR
10F3       JMP   MAKCN         C3EB10 TRY AGIAN
10F6 EXBYT ORA   A             B7     SET FLAG IF A SUBTRACTED TO ZERO
10F7       RLC                 07     DOUBLE VALUE IN A
10F8       MOV   B     A       47     SAVE A IN B
10F9       MOV   A     M       7E     GET BYTE OF OPCOUNT INFORMATION
10FA       JZ    FSTCN         CA0211 EXTRACT DATA IF Z FLAG=1
10FD RLC10 RLC                 07     ROTATE ACCUMULATOR LEFT ONCE
10FE       DCR   B             05     ONE LESS FOR B
10FF       JNZ   RLC10         C2FD10 ROTATE SOME MORE B NOT ZERO YET
1102 FSTCN RLC                 07     ROTATE VALUE TO BIT 0
1103       RLC                 07     AND TO BIT 1
1104       ANI   03            E603   MASK OUT REST OF BYTE
1106       RET                 C9     OPCODE BYTE COUNT IS NOW IN A
****************************************************************
*       CHECK TABLE FOR BYTE AND JUMP TO CORRESPONDING ADDRESS  *
****************************************************************
1107 TBCHK INR   B             04     INCREMENT LOOP COUNTER
1108 TBC1  DCR   B             05     DECREMENT LOOP COUNTER
1109       RZ                  C8     IF B IS ZERO, BYTE NOT IN TABLE
110A       CMP   M             BE     SEE IF BYTE IN TABLE
110B       JZ    TBC2          CA1411 IF SO, THEN NEED ADDRESS
110E       INX   H             23     SKIP DATA BYTE
110F       INX   H             23     AND SKIP THE TWO
1110       INX   H             23     SUBSEQUENT ADDRESS BYTES
1111       JMP   TBC1          C30811 DO THE LOOP AGAIN
1114 TBC2  POP   D             D1     RESTORE STACK POINTER
1115       INX   H             23     POINT TO LOW ORDER ADDRESS BYTE
1116       JMP   HLJMP         C32611 GET ADDRESS AND MAKE JUMP
****************************************************************
*       SYSTEM TYPE OPCODES--HIGH ORDER OCTAL PART IS 3         *
*       A JUMP IS MADE TO THE APPROPRIATE ROUTINE CONSISTENT    *
*       WITH THE FOLLOWING OCTAL ORGANIAZATION OF THE OPCODE    *
*                                                               *
*    RETYP. EXCYC . BRTYP  .EXCYC . CALTY . EXCYC  . EXCYC .RSTTY *
*    ..........................................................  *
*     3X0 .  3X1  .  3X2   . 3X3  . 3X4   .  3X5   .  3X6  . 3X7 *
*    ..........................................................  *
*     RNZ . POP B .  JNZ   . JMP  . CNZ   . PUSH B .  ADI  . RST0 *
*     RZ  . RET   .  JZ    . ---  . CZ    . CALL   .  ACI  . RST1 *
*     RNC . ---   .  JNC   . OUT  . CNC   . PUSH D .  SUI  . RST2 *
*     RC  . POPH  .  JC    . IN   . CC    . ---    .  SBI  . RST3 *
*     RPO . PCHL  .  JPO   . XTHL . CPO   . PUSH H .  ANI  . RST4 *
*     RPE . POP H .  JPE   . XCHG . CPE   . ---    .  XRI  . RST5 *
*     RP  . POP P .  JP    . DI   . CP    . PUSH P .  ORI  . RST6 *
*     RM  . SPHL  .  JM    . EI   . CM    . ---    .  CPI  . RST7 *
*                                                               *
*       NOTE:                                                   *
*              JMP, RET, CALL, AND PCHL HAVE BEEN ELIMINATED    *
*              FROM CONSIDERATION AT THIS POINT                 *
****************************************************************
1119 SYSOP LXI   H     SYTBL   211510 GET START OF SYTBL ADDRESS
111C       LDA   EXOP          3AD01F GET OPCODE
111F       ANI   07            E607   GET LOW ORDER OCTAL PART
****************************************************************
*              INDIRECT JUMP COMPUTE ROUTINE                    *
****************************************************************
1121 INJMP ADD   A             87     MULTIPLY BY TWO
1122       MOV   E     A       5F     SET UP FOR COMPUTING
1123       MVI   D     20      1600   THE OFFSET
1125       DAD   D             19     COMPUTE THE OFFSET
1126 HLJMP MOV   E     M       5E     LOW ORDER ADDRESS
1127       INX   H             23     POINT TO HIGH ORDER ADDRESS
1128       MOV   D     M       56     HIGH ORDER ADDRES
1129       XCHG                EB     MOVE TO H-L
112A       PCHL                E9     MAKE THE JUMP
****************************************************************
*              INSTRUCTION SIMULATION ROUTINES                  *
****************************************************************
112B PCH   LHLD  LSAVE         2ADE1F GET ORIGINAL VALUE OF H-L
112E       SHLD  PSAVE         22E11F UPDATE PROGRAM COUNTER
1131       JMP   FINI          C37D10 FINISHED
1134 BRTYP LHLD  EXOP1         2AD11F GET USER'S JUMP-TO ADDRESS
1137       XCHG                EB     PASS IN D-E
1138       LXI   H     JMPSC   213E11 SUCCESSFUL JUMP ADDRESS
113B       JMP   BRSB          C34E11 COMPLETE BRANCH OPERATION
113E JMPSC LHLD  MSTC2         2AD61F GET JUMP ADDRESS FROM MSTC2
1141       SHLD  PSAVE         22E11F STORE THE ADDRESS
1144       JMP   FINI          C37D10 FINISHED
1147 CALTY LHLD  EXOP1         2AD11F GET USER'S CALL ADDRESS
114A       XCHG                EB     PASS IN D-E
114B       LXI   H     CALSC   217511 SUCCESSFUL CALL ADDRESS
114E BRSB  SHLD  EXOP1         22D11F REPLACE JUMP ADDRESS IN RAM
1151       XCHG                EB     PASS USER'S SAVED ADDRESS IN H-L
1152       JMP   FLRS          C35811 RESTORE FLAGS AND EXECUTE
1155 RETYP LXI   H     RETSC   216411 RETURN SUCCESSFUL ADDRESS
1158 FLRS  POP   PSW           F1     RESTORE FLAGS
1159       PUSH  PSW           F5     RESTORE STACK POINTER TO MSTCK
115A       PUSH  H             E5     PUT USER'S ADDRESS AT MSTCK-2
115B       LXI   H     FINI    217D10 GET RETURN POINT
115E       SHLD  EXOP4         22D41F FOR UNSUCCESSFUL CALL/RET
1161       JMP   EXOP          C3D01F EXECUTION STEP
1164 RETSC LHLD  SSAVE         2ADF1F GET ORIGINAL STACK POINTER VALUE
1167       SPHL                F9     RESET STACK POINTER TO THIS VALUE
1168       POP   H             E1     RETURN ADDRESS - ORIGINAL STACK
1169       JMP   SAVPS         C38011 UPDATE PSAVE AND SSAVE
116C RSTTY LDA   EXOP          3AD01F GET OPCODE
116F       ANI   38            E638   EXTRACT MIDDLE ORDER OCTAL PART
1171       MOV   L     A       6F     WHICH IS LOW ORDER ADDRESS
1172       MVI   H     00      2600   FILL H WITH 0'S
1174       PUSH  H             E5     SAVE AT MSTCK-2
1175 CALSC LHLD  SSAVE         2ADF1F GET ORIGINAL STACK POINTER VALUE
1178       SPHL                F9     RESET STACK POINTER TO THIS VALUE
1179       LHLD  PSAVE         2AE11F GET RETURN ADDRESS AFTER CALL
117C       PUSH  H             E5     PUT ON ORIGINAL STACK
117D       LHLD  MSTC2         2AD61F GET ORIGINAL CALL ADDRESS
1180 SAVPS SHLD  PSAVE         22E11F UPDATE PSAVE TO CORRECT ADDRESS
1183       LXI   H     0000    210000 FILL H-L WITH 0'S
1186       DAD   SP            39     GET VALUE OF STACK POINTER
1187       SHLD  SSAVE         22DF1F SAVE - NEW VALUE OF STACK POINTER
```

tion. Furthermore, all these branch type instructions have about eight other variations depending on flag conditions. So we must:

1. Save the original address.
2. Substitute our own address.
3. Restore the flags.
4. Decide whether we branched or not, and take the appropriate action.

The appropriate action is:

1. Update PSAVE depending on whether the branch occurred or not.
2. Update SSAVE depending on whether a stack operation occurred or not.

Note that the instruction PCHL (replace the program counter with value in H and L) needs special treatment. A halt instruction should not be executed!

This necessitates a lot of branching around depending on what we have to do. The code may take some study to understand. The entry point is at 1000 hexadecimal (all numbers are in hexadecimal).

Upon entry it is assumed that PSAVE is set to the starting address. Space is provided for a jump to a user-defined display routine and to return to the user's monitor. The reentry point is at SSTEP. The ORG for the programmable memory execution area allows about 30 bytes of an auxiliary storage area after the register save area and before the end of the 1 K byte block of memory. In practice, the setting of MSTCK controls the register storage location.

Author's Note:

The program in listing 1 has been written for use on the 8080 microprocessor based Hewlett-Packard 2649A terminal. This is to be used in conjunction with a disk storage system as the controller for 40 point of sale cash registers of a large department store chain. That is my professional interest.

This code was adapted from the original (and well debugged) source code. It should work fine. It is in a somewhat nonstandard Intel 8080 assembly code format. There are a few abbreviations and fixed format. Mostly I wanted to present some of the concepts that are involved.

I have an IMSAI which I am configuring from the ground up. The whole monitor with display, change, go, single stepping and breakpoint commands takes about 1 K bytes of memory.■

Solution to Machine Language Puzzler (see page 52)

The subroutine calls itself, and an infinite loop is set up that is exactly one instruction in length. Each time the CALL is executed, however, a return address of 0000 is pushed onto the stack. As a result of the LXI SP, FFFD instruction, the stack area will begin immediately adjacent in address to the CALL (recall that PUSH causes stack data to enter one address below that of the current stack pointer). Furthermore, the stack will grow away from the address of the CALL instruction.

At first, the LXI instruction code will be replaced by 0s, which is the value for each byte of the CALL's return address. As time progresses, two bytes at a time are cleared as 0000 return addresses are repeatedly pushed.

Eventually, all memory lower in address than the CALL will be cleared. When the stack attempts to grow below address 0000, it will wrap around to the top of memory and begin to affect the CALL instruction code. Since the last byte of each push enters an *odd* address, a byte pair constituting the 0000 return address will enter at hexadecimal addresses 0000 and FFFF, respectively. This action changes the high byte of the CALL address.

The next CALL will then be to address 00FD. Since 0s have been written in all other memory, the processor will execute NOP instructions until address FFFD, when one final CALL will be executed. This time, the last two remaining bytes of nonzero code (the CALL instruction code) are overwritten with zeros. Henceforth, starting at address FFFD, the processor will cycle through memory in NOPs, forever. In short, all memory is cleared and the processor hangs up.

The same operation occurs as in part (a), except when the NOP instructions are executed. We assume, as is typical of most machines, that uninstalled memory returns hexadecimal FF when addressed. In this case, FFFD and FFFE are not cleared, because FF, when executed, causes a RST 7. Now 0101 as a return address to RST 7 is pushed repeatedly, and eventually LXI B, 0101 is executed forever in a loop between hexadecimal addresses 0000 and 0102.■

Elements of Statistical Computation

Alan B Forsythe, PhD
University of California
Health Services Computing Facility
Los Angeles CA 90024

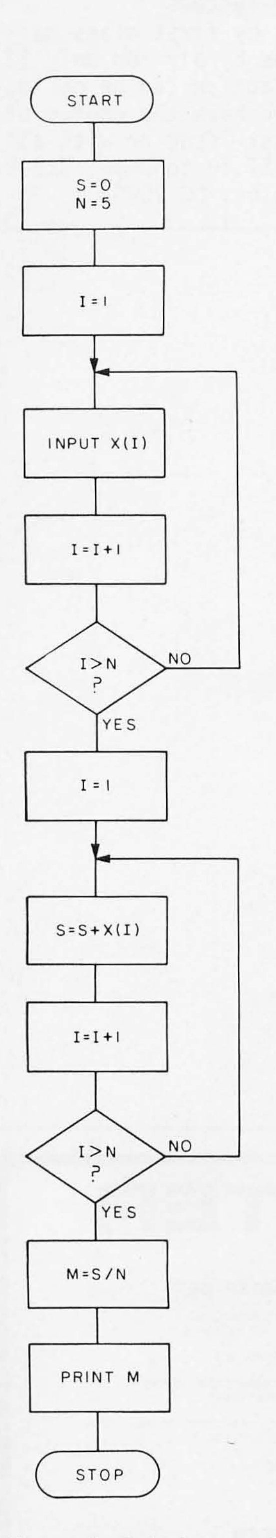


Figure 1: Calculation of the mean of a set of data. Using this method, the data values are used twice: once for input and the second time for summation. The method is wasteful of both time and memory.

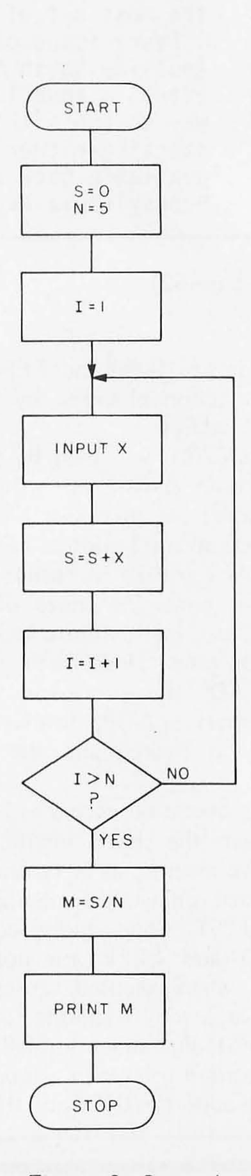


Figure 2: Second method of calculating the mean of a set of data. The data is examined only once and it does not need to be stored.

The *mean* (average) is a frequently used statistic to summarize a set of data. It is used to report a *typical* value. For example, suppose we surveyed five automobile dealers and asked the price of the super luxury model car. The prices quoted for the car are:

Dealer	Price
1	$48,499
2	$48,503
3	$48,500
4	$48,498
5	$48,500

We could report the average price as:

$$\frac{(48499+48503+48500+48498+48500)}{5}$$

The sample mean price is $48,500.

The mean is defined as the sum of the values divided by the number of values. This can be written using the symbol Σ (the Greek capital sigma) to stand for the result of summation. The average of N values is:

$$\text{mean} = \frac{1}{N}\Sigma X$$

The sigma indicates a *sum* over all the data points.

Let's look at two algorithmic flowcharts for the calculation of the mean. The first, figure 1, reads all the data and stores it in array X. Next, it sums the values and performs the division. The second flowchart, figure 2, takes advantage of the fact that we really do not need to keep the values in memory to form the average. Each value is added to the total sum as it is entered. The program loops through the data once rather than twice. The second method is more efficient because it uses less memory, is faster, and uses fewer instructions. This difference in efficiency may not be critical when only five values are being averaged, but when many observations are involved, the importance increases.

```
10    REM FIRST PROGRAM TO CALCULATE THE MEAN
20    DIM X(5)
30    LET N=5
40    LET S=0
50    FOR I=1 TO N
60    INPUT X(I)
70    NEXT I
80    FOR I=1 TO N
90    LET S=S+X(I)
100   NEXT I
110   LET M=S/N
120   PRINT "THE MEAN IS "; M
130   STOP

*RUN
?48499
?48503
?48500
?48498
?48500
THE MEAN IS 48500
```

Listing 1: BASIC program to perform the algorithm given in figure 1. Note the storage of data in array X.

```
10    REM SECOND PROGRAM TO CALCULATE THE MEAN
30    LET N = 5
40    LET S = 0
50    FOR I=1 TO N
60    INPUT X
65    LET S = S+X
70    NEXT I
110   LET M = S/N
120   PRINT "THE MEAN IS "; M
130   STOP

*RUN
?48499
?48503
?48500
?48498
?48500
THE MEAN IS 48500
```

Listing 2: BASIC program to perform the algorithm given in figure 2. It is faster than the program in listing 1, since it uses the data only once. It also uses less memory space since the data is not stored.

There is another difference worth mentioning. A noticeable pause between the entry of the last value and the printing of the average may result when the program in listing 1 is run, since the calculations are concentrated at this point in the program. Since the program in listing 2 does the summing after each data entry, the effect is a quick final response. Of these two methods, one is more efficient even though both would give the same answer.

There is another statistical calculation in which we can easily obtain the wrong answer unless we are careful. The average of a set of values gives only part of the picture. You may have heard of the man who drowned in a river that had an average depth of two feet. He happened to be in a section that was 12 feet deep. The moral is that we often want to know how much the values vary from the mean. This is commonly reported as the standard deviation. The usual symbol for the mean is $\overline{X}$ (read this as "X bar") and the standard deviation is abbreviated as the letter s. It is also convenient to have a name for s^2. It is called the variance, and its definition is:

$$s^2 = \frac{1}{N-1}\Sigma\,(X-\overline{X})^2$$

where N is the total number of samples taken. It follows, then, that the standard deviation is:

$$s = \sqrt{\frac{1}{N-1}\Sigma\,(X-\overline{X})^2}$$

The sigma (Σ) once again tells us that we are to perform a sum. This time we sum the deviations from the mean after we have squared them. That is, for each value we first subtract the mean. Next, we multiply the difference by itself to form its square. We add these squared differences for all the values and then divide by N−1. The last step to obtain the standard deviation is to take the square root of s^2. It is calculations like these that make statisticians appreciate computers.

Before we move on to programs for the calculation of the standard deviation, let us work through these calculations for our example automobile prices. Remember that the average price was $48,500. The price from the first dealer was $48,499. The first dealer deviates from the mean by (49,499−48,500) or −1 dollar. The square of this deviation is 1. That is the first part of our sum. The squared deviation from the average for the second dealer is 9, and so on. The sum for all five dealers is 14. When we divide by 4 and then take the square root, we obtain a standard deviation of 1.87. The smaller this value, the closer the set of observations are to their mean. A large standard deviation tells us the data varied greatly from observation to observation. A standard deviation of zero says that all the observed values were equal to each other.

If you were to go to a statistics book you would probably find another formula for this calculation which is sometimes called the computing formula. This method of calculation uses the fact that $\Sigma\,(X-\overline{X})^2$ is algebraically identical to $\Sigma X^2 - N\overline{X}^2$. This tells us that we can calculate the sum of the squares of the individual values without sub-

tracting the mean each time. We can multiply the square of the mean by N and subtract.

Mathematically these two formulas give the same answer. *Numerically* they are different. Why? They differ because our computers do not store or calculate numbers perfectly. If each number is stored in our computer in four 8 bit bytes, or 32 bits, we have about 7 digit accuracy for any single step in our calculation. The second formula lets the errors accumulate. This fact is not considered in the mathematical proof.

```
10    REM POOR PROGRAM FOR CALCULATING THE
15    REM MEAN AND STANDARD DEVIATION
20    LET N=5
30    LET S=0
40    LET S2=0
50    FOR I=1 TO N
60    INPUT X
70    LET S=S+X
80    LET S2=S2+X*X
90    NEXT I
100   LET M=S/N
110   PRINT "THE MEAN IS "; M
120   LET D=N*M*M
130   LET V=(S2-D)/(N-1)
140   LET S=SQRT(V)
150   PRINT "THE STANDARD DEVIATION IS "; S
160   STOP

*RUN
?48499
?48503
?48500
?48498
?48500
THE MEAN IS 48500
THE STANDARD DEVIATION IS 0
```

Listing 3: Incorrect program for calculating the mean and standard deviation of a set of data. This program correctly calculates the mean but arrives at an incorrect value for the standard deviation.

```
10    REM REASONABLE PROGRAM TO CALCULATE
15    REM MEAN AND STANDARD DEVIATION
20    LET N=5
30    LET S=0
40    LET S2=0
50    FOR I=1 TO N
60    INPUT X
65    LET D=X-S
70    LET S=S+D/I
80    LET S2=S2+D*(X-S)
90    NEXT I
110   PRINT "THE MEAN IS" ; S
130   LET V=S2/(N-1)
140   LET S=SQRT(V)
150   PRINT "THE STANDARD DEVIATION IS "; S
160   STOP
*RUN
?48499
?48503
?48500
?48498
?48500
THE MEAN IS 48500
THE STANDARD DEVIATION IS 1.87083
```

Listing 4: Accurate program for calculating the mean and standard deviation.

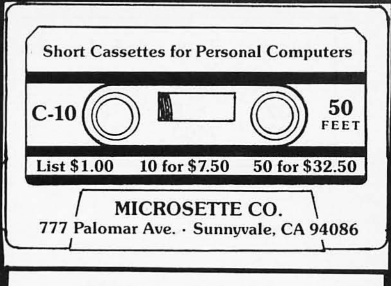
Why is this second form given in books? First of all, some calculators work with more digits than many personal computers. Secondly, it is much easier and faster than using the formula for the definition. The Hewlett-Packard HP-21 calculator shows 10 decimal digits, as does the Texas Instruments SR-52. They can afford the luxury of roundoff error that seriously hurts those with 32 bits. Before we show how we can almost get the best of both worlds, let us look at the numerical results of using this calculating formula on computers in which each number is kept in four bytes. (The IBM 370 single precision and many microcomputer BASICs are just two common examples.)

The program in listing 3 correctly gives the mean as 48,500. The standard deviation is computed to be 0. This is incorrect. All the prices are not equal to 48,500. Several books of BASIC programs include the calculation of the standard deviation. Those I checked out give the wrong answer for this set of data.

We know that we get accurate results if we first calculate the mean and use the differences from it in our calculation of the standard deviation. But we would like to avoid keeping all the data values in memory. Another alternative is to enter the data twice, but this seems unreasonable. A compromise between the two calculating forms that is not very sensitive to the accumulation of roundoff errors is available. The principle is to calculate a provisional mean as each value is entered, and to square the deviations from this mean.

The program in listing 4 reports the standard deviation for this set of data correctly as 1.87. This program is not as obvious as it may look. Notice that line 70 has a division by I and not N in the calculation of the mean. Line 80 also has a little trick. The product of D and X–S is not D*D, because S has been changed since D was last calculated. If you have trouble working through the algebra, put print statements after lines 65, 70 and 80.

While writing statistical programs for microcomputers, I have been reminded of several important facts I hope you will keep in mind. Good programming takes thought. We must balance several factors: speed, accuracy and memory requirements. Algebra is an essential tool, but we must remember that our machines have finite precision and adjust our program accordingly. A small machine can be quite potent, given adequate software.■



Duplication Services

Microsette also offers professional duplication services for Commodore PET and Radio Shack TRS-80 Level I and Level II cassettes. Our service provides mastering, quality control, all material including two-piece box, affixing of your labels or supplying our blank labels and shipping. Prices start at $1.00 each in 100 quantity.

MICROSETTE CO.
777 Palomar Ave. · Sunnyvale, CA 94086

Circle 229 on inquiry card.

TRS - 80

SPECIAL PROMOTION SALE

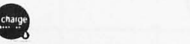
SAVE 10%, 15% or more on **ALL** Computers, Peripherals, Software, and **ALL** other fine Radio Shack® products.

NO TAXES on out-of-state shipments.

FREE Surface delivery in U.S.

WARRANTIES will be honored by your local Radio Shack® store.

Offered exclusively by
Radio Shack®
Authorized Sales Center
1117 Conway
Mission, Texas 78572
(512) 585-2765



Circle 304 on inquiry card.

tek

- Sanyo monitor model VM4209 . . . . . .$109.95
- ASR 33 Teletypes Used — Good conditions $395.95
- ADM-3A Terminal kit. . . .$549.99

All items shipped with warranty and are postpaid. Please allow 4-6 weeks for delivery. Florida residents add 4% sales tax.
Send check or money order to:

tek peripherals
5110 S.W. 106 Avenue
Miami, FL 33165

Circle 349 on inquiry card.

Z80 ASSEMBLER PACKAGE

for North Star HORIZON using full set of ZILOG mnemonics.

(1) EDITOR — generates and edits source code with comments
(2) ASSEMBLER — generates opcode and links subroutines
(3) LOADER — generates and loads binary code

Short or long programs can be processed by the file oriented package written in North Star BASIC (Release 4). File handling, string insertion and deletion and table searches are implemented in Z80 machine code for a fast operation.

Diskette and manual with full program listing in BASIC and Z80 commented assembler:

$35.

NEMCO DATA PROCESSING
9 WALNUT STREET
RUTHERFORD, NJ 07070

Circle 269 on inquiry card.

PROGRAMMING CONTEST !

MICRO–PUZZLE NO. 2

1st PRIZE– FLOPPY DISK SYSTEM OR $1000 CASH
2nd PRIZE– FLOATING POINT BOARD OR $400 CASH
3rd PRIZE– AUDIO INTERFACE OR $200 CASH

M	I	C	R	O	16	
O	N	L	A	P	11	
O	L	I	V	E	14	= 61
N	E	V	E	N	9	
S	T	E	D	S	11	

WHY CAN THE HUMAN MIND SOLVE THIS PUZZLE EASIER THAN THE MACHINE MIND? SOMEWHERE ON EARTH THERE MUST BE A COMPUTER THAT CAN BEAT THE HUMANS WHO ENTER THIS CONTEST. THIS IS A TOTALLY OBJECTIVE PUZZLE. PERFECT FOR COMPUTER SOLUTION. WINNERS ARE DETERMINED SOLELY ON THEIR ABILITY TO GET THE HIGHEST SCORE IN ABOUT 1 MONTHS TIME. 4th & 5th PRIZE ARE NOW $100.

For entry form and official rules send a self addressed stamped envelope to:
MICRO-PUZZLES Dept. A 7858 Cantaloupe Ave.
Van Nuys, California 91402

Circle 227 on inquiry card.

MICRO-VERTER

A SPECIAL COLOR MODULATOR FOR APPLE II USERS!!

UHF Version. Operates above channel 14. Eliminates worms!

Operates above the switching harmonics of the computer, thereby yielding a cleaner, worm-free picture. Tunable over a minimum of 4 channels. Interfaces directly with the Apple II as well as most other micros. Comes with video cable and RF output stub coupler. Two-toned cowl type decorator cabinet. Size: 5.5 cm x 8.5 cm x 11.5 cm. Power: +5V. Current approx. 1 ma. Self-powered with 4 pencell batteries. Operating life in excess of 1000 hours or near shelf-life of batteries. Excellent stability. Precise frequency adjustment. No assembly required except for installation of batteries, not supplied. MODEL MVX-500.

AVAILABLE FROM YOUR LOCAL COMPUTER DEALER or direct from ATV RESEARCH. COST including shipping anywhere in USA and Canada — $35.00.

"PIXE-PLEXER" An IC type video-to-RF modulator includes FM sound sub-carrier, color subcarrier and separate R-Y and B-Y inputs. Designed around the LM-1889 chip. A designer's dream with full data sheets. Model PXP-4500. Kit form. $24.50 postpaid.

"PIXE-VERTER" The original computer video-to-RF interface module. Kit form: $8.50 Model PXV-2A

PHONE or WRITE TODAY. DIAL 402-987-3771.

13-B Broadway **ATV Research** Dakota City, Nebr. 68731

Circle 22 on inquiry card.

MICRO SOFTWARE DEVELOPMENT

THE COMPANY: Fastest growing commercial systems manufacturer in U.S. 1978 sales in excess of 150 million.

DUTIES: Design, code & debug 8080 assembly programs for interactive graphics, text editing, diagnostics, and other applications.

REQUIREMENTS: One or more years assembly development exp. with any hardware. Degree a plus.

COMPENSATION: 16.5K-28K. Excellent benefits.

GROWTH: Company has promoted over 50% of employees in last 4 years.

CONTACT: Dave Adams, (617) 246-2815. N.E. Recruiters, 6 Lakeside Office Park, Wakefield, MA 01880.

All fees, relocation and interviewing expenses assumed by company.

OMSI PASCAL-1™

All PDP-11s $1500

RSTS/E, RSX-11, RT-11, and IAS



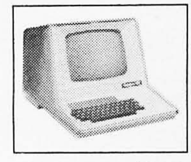
Oregon minicomputer Software inc
TWX 910-464-4779
2340 SW Canyon Road
Portland, Oregon 97201
(503) 226-7760

Circle 291 on inquiry card.

HAZELTINE 1400

only

$649.95!

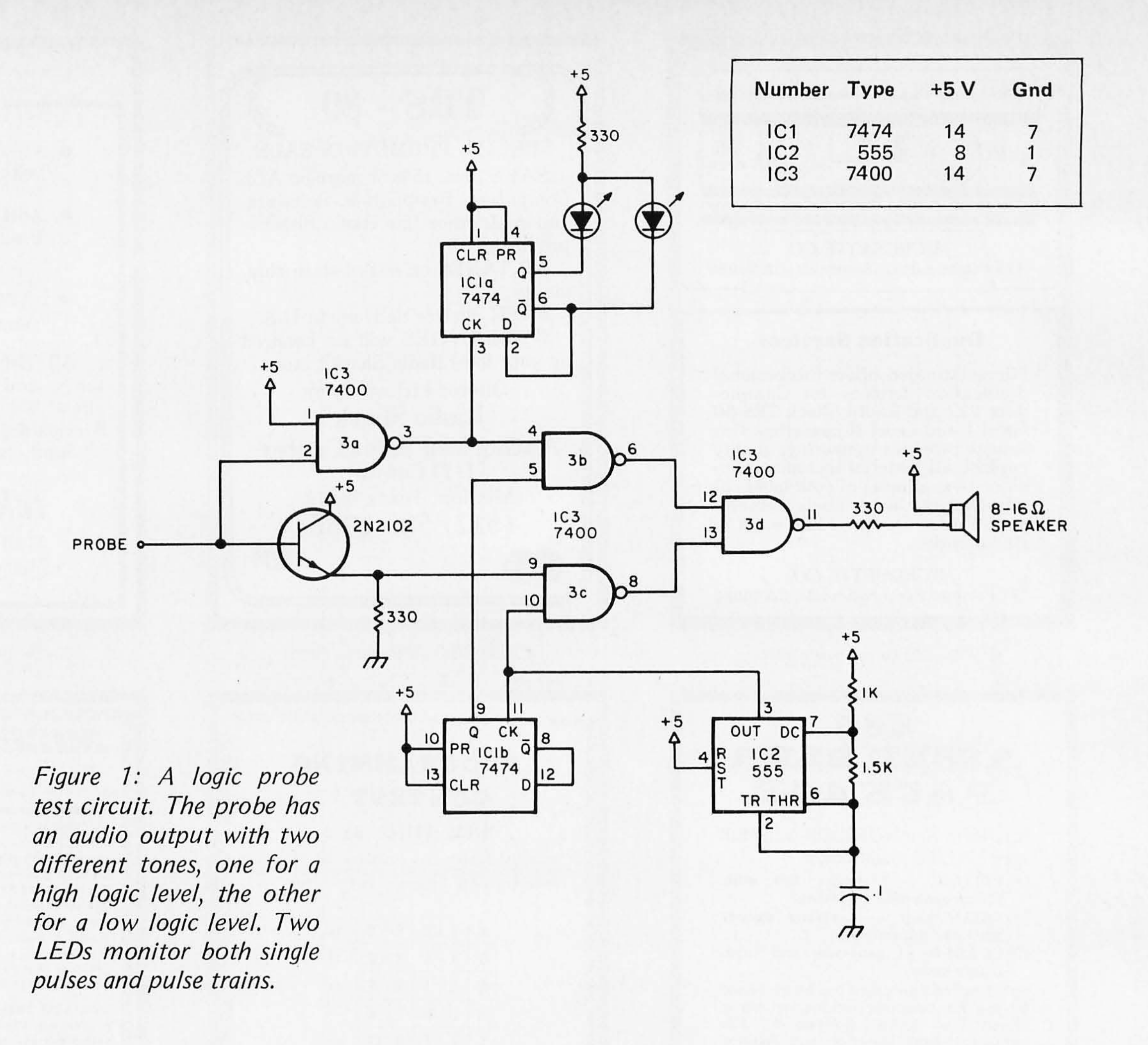


- Verbatim Mini Diskettes $3.70 each (boxes of 10)
- Two-tier walnut formica enclosure for SA-400 Shugart. . . $39.95
- Typewriter Ribbons (many makes such as Diablo, Centronics, DEC and print wheels)
- North Star check balancing program. . .$50.00
- Centronics 779 tractor — $950
- Horizon II ass. $1000 — $1999.00

Mail Order Only.

TORA SYSTEM
29-02 23rd Avenue
Astoria NY 11105
(212) 932-3533

Circle 346 on inquiry card.



Number	Type	+5 V	Gnd
IC1	7474	14	7
IC2	555	8	1
IC3	7400	14	7

Figure 1: A logic probe test circuit. The probe has an audio output with two different tones, one for a high logic level, the other for a low logic level. Two LEDs monitor both single pulses and pulse trains.

An Audible Logic Test Probe

James L Woodward

Most logic probes require the placement of a point on the lead to be checked and the surveillance of a light emitting diode (LED) indicator. In a well-lit room, this is sometimes difficult to interpret and often requires three hands. The probe offered here has an audio output that is off when the probe is floating, emits a low tone for a low state, and a tone one octave higher for a high state. Pulse trains and single pulses are monitored with a pair of LEDs, and the probe can be anything from a rough length of wire to a lead clip; none of the values are critical.

Construction

For use as a simple state checker, the construction is completely noncritical. However, if the circuit is to be used to measure high frequency circuits, it should be built using good technique, with plenty of bypass capacitors on a good quality board. Note that there are audio frequencies to be bypassed as well as high frequency information (a couple of capacitors in the microfarad range across the power supply would be wise). Practically any NPN transistors will do in the circuit of figure 1.

Theory

The timer, IC2, is wired to produce a pleasing tone frequency for the high state. This is divided by two by IC1b for the octave-lower low output. With no voltage on the probe, the 330 Ω resistor on the emitter of the transistor holds pin 8 of IC3c high, despite the audio on the other input. Similarly, pin 2 of IC3a floats in a high state. The signal is inverted by IC3a and holds the output of IC3b high. With both inputs high, IC3d is low and, while a small DC current flows through the speaker, no sound is produced. A high level on the probe has no effect on IC3a and IC3b, but allows the

transistor to conduct, thus forcing pin 9 of IC3c high and allowing the audio frequency through in inverted form (the inversion is irrelevant in this application). Since the other input of IC3d is high, the audio goes straight through to the speaker. A low on the input probe produces the complementary effect; being inverted by IC3a, it allows IC3b to output the low frequency from the divider, which goes through IC3d and the speaker.

Having half of a 7474 flip flop available, I decided to add a simple pulse catcher and pulse train monitor. IC1a is also wired as a divide by two, with both outputs going to LEDs. A single pulse will change the states of the two LEDs (as will a loose touch on the probe, so better use a clip for this), while a train will light them both. Technically, the duty cycle of the pulse train affects the relative brightness of the two lamps and the combination of the two audio frequencies, but I doubt that this is useful in a real case. However, the audio frequency shift is a rapid indication of the presence of a non-steady state. The logic of the divide by two pulse catcher may be extended by cascading several counters to divide the frequency of a fast pulse train down to the point where it could be mounted by eye, stopwatch, or frequency counter.■

Event Queue

In order to gain optimum coverage of your organization's computer conferences, seminars, workshops, courses, etc, notice should reach our office at least three months in advance of the date of the event. Entries should be sent to: Event Queue, BYTE Publications Inc, 70 Main St, Peterborough NH 03458. Each month we publish the current contents of the queue for the month of the cover date and the two following calendar months. Thus a given event may appear as many as three times in this section if it is sent to us far enough in advance.

January 8 & 9, The First International Symposium on Mini and Microcomputers in Control, Islandia Hyatt House, San Diego CA. The symposium is sponsored by The International Society for Mini and Microcomputers (ISMM) and IEEE Control Systems Society. Over 50 papers are to be presented including tutorials and surveys. Contact Computers in Control Symposium, POB 2481, Anaheim CA 92804, (714) 774-6144.

January 8-12, Structured Programming and Software Engineering, George Washington University, Washington DC. This course is designed for experienced program architects, designers and managers. It will provide up to date technical knowledge of logical expression, analysis and invention for performing and managing software architecture, design and production. Presentations will cover principles and applications in structured programming and software engineering. Design workshops with analysis and review sessions will provide actual practice in problem solving. Contact George Washington University, Continuing Engineering Education, Washington DC 20052.

January 11-13, Software Engineering for Mini/Microcomputer Systems Seminar, Airport Marina Hotel, Los Angeles CA. Polytechnic Institute of New York and the Institute for Advanced Professional Studies are presenting a 3 day seminar for hardware designers and programmers. This seminar will cover computer system design concepts within the context of the operation and application of the LSI-11. Contact Prof Donald D French, Institute for Advanced Professional Studies, 1 Gateway Ctr, Newton MA 02158, (617) 964-1412.

January 12-14, PerBizComp '79, Washington Hilton Hotel, Washington DC. Business and personal microcomputer show. Contact Felsburg Associates Inc, 12203 Raritan Ln, POB 735, Bowie MD 20715, (301) 262-0305.

January 15-17, Bit-Slice Microcomputer and Digital System Design Seminar, Airport Marina Hotel, Los Angeles CA. Polytechnic Institute of New York and the Institute for Advanced Professional Studies are presenting this 3 day seminar for digital systems engineers. This seminar will cover review of computer architecture and organization, design principles for bit-slice digital systems, introduction to microprogramming, detailed study of the 2900 bit-slice family and comparison with similar hardware, survey of firmware aids and support tools for system development, an introduction to emulation techniques, and trends in bit-slice digital system design. Contact Prof Donald D French, Institute for Advanced Professional Studies, 1 Gateway Ctr, Newton MA 02158, (617) 964-1412.

January 15-17, Minicomputers and Distributed Processing, San Francisco. This 3 day seminar will examine the uses, economics, programming and implementation of minicomputers. Contact Philip M Kowlen, director, Center for Continuing Education, The University of Chicago, 1307 E 60th St, Chicago IL 60637.

January 16-18, The Seventh International Symposium on Mini and Microcomputers, Disneyland Hotel, Anaheim CA. The symposium is sponsored by The International Society for Mini and Microcomputers. The symposium will cover all aspects of mini and microcomputers and their applications. Over 60 papers are to be presented. Contact MIMI'79 Anaheim, POB 2481, Anaheim CA 92804, (714) 774-6144.

January 17-19, Distributed Minicomputer Networks, Ramada Inn, Arlington VA. This seminar will address the minicomputer from the viewpoint of the distributed network user. The structure and management of a large data base and software problems with the tradeoffs of languages utilized, hardware types, input and output options, device controllers, system failure and recovery, sample application case studies and the economics of minicomputer applications will be covered in depth. Contact The Institute for Professional Education, Suite 601, 1901 N Fort Myer Dr, Arlington VA 22209, (703) 527-8700.

January 24-27, International Micro-

computers/Minicomputers Microprocessors '79/Japan, Harumi Exhibition Center, Tokyo. Contact ISCM, 222 W Adams St, Chicago IL 60606, (312) 263-4866.

January 29-31, 27th Annual Physics Show, New York Hilton Hotel, New York. This show is to be held in conjunction with the combined annual meeting of the American Physical Society and the American Association of Physics Teachers. Contact Ed Greeley, American Institute of Physics, 335 E 45th St, New York NY 10017.

January 30-February 1, Communication Networks Conference and Exposition, Sheraton Park Hotel, Washington DC. Designed to bring together communication network users, consultants, vendors and regulatory officials so that issues can be discussed and analyzed. It is particularly aimed at executives and managers who purchase communication products and services. Contact The Conference Company, 60 Austin St, Newton MA 02160.

February 1-3, Microprocessor Programming Workshop With a Take-Home Microprocessor, Jefferson Plaza Building, Arlington VA. Sponsored by the IEEE, this 3 day workshop is intended for the practicing engineer, engineering manager and programmer. The course objective is to provide state of the art information in order to acquire an understanding of the place of microprocessors as replacements for wired logic and as controllers, to provide the capability of understanding the design of systems involving microprocessors, and the ability to program the Motorola M6800 microprocessor in machine language. All students will have their own microprocessors and laboratory equipment. Contact IEEE Service Ctr, 2145 Hoes Lane, Piscataway NJ 08854.

February 13-15, The National Office Exhibition and Conference, Harbour Castle Hilton Convention Center, Toronto Ontario. This 3 day exhibition will provide a showplace for approximately 100 exhibitors in the areas of word processing, office computers, office equipment and furniture. Contact *Canadian Office* Magazine, 2 Bloor St W, Suite 2504, Toronto Ontario, CANADA M4W 3E2, (416) 967-6200.

February 14-16, The IEEE International Solid-State Circuits Conference, Philadelphia PA. Forum for the presentation of new advancements in all aspects of solid-state circuits. Contact Lewis Winner, 301 Almeria Av, POB 343788, Coral Gables FL 33134.

March 19-20, Microcomputers: Operating Principles, Hardware and Software Seminar, Holiday Inn, Palo Alto CA. Polytechnic Institute of New York and the Institute for Advanced Professional Studies are presenting this two day seminar for engineers, programmers, and technical managers involved with selection of microprocessors and design of microprocessor based systems. The seminar will cover the underlying concepts governing microprocessor operation, architecture, and systems design. Microcomputer elements and their interrelationships will be discussed, emphasizing features important in determining whether a particular microcomputer will be suitable for a given task. Contact Prof Donald D French, Institute for Advanced Professional Studies, 1 Gateway Ctr, Newton MA 02158, (617) 964-1412.

March 19-21, Modern Integrated Circuits, George Washington University, Washington DC. This course is structured to meet the needs of engineers, scientists and technical managers who desire a better understanding of the latest technological advances in the area of integrated circuits. As such it examines all aspects of integrated circuit technology, starting from fundamental principles of construction and operation, to the most recent devices, their characteristics and specifications. A significant part of the course deals with the application of integrated circuits in linear and digital systems. Specific topics to be covered include detailed design examples of circuits using operational amplifiers and active filters, as well as computer arithmetic units, registers and memories. Contact George Washington University, Continuing Engineering Education Program, Washington DC 20052.

March 21-23, Microcomputer Hardware and System Design Seminar, Holiday Inn, Palo Alto CA. Polytechnic Institute of New York and the Institute for Advanced Professional Studies are presenting this three day seminar for engineers, programmers and technical managers with a working knowledge of digital hardware design and familiarity with the underlying concepts governing microprocessor operation, architecture and systems design. This seminar will cover the operation, architecture, instruction set and design techniques for 8 bit microprocessors. The spectrum of applications from data processing to control will be illustrated with fully developed case studies. Contact Prof Donald D French, Institute for Advanced Professional Studies, 1 Gateway Ctr, Newton MA 02158, (617) 964-1412.

March 22-24, Future Fair, Memorial Coliseum, Portland OR. This northwest regional exposition will feature both professional and personal data processing products and services. Contact WES/COM, POB 4047, Portland OR 97208.

March 25-28, Expo '79, Los Angeles Marriott, Los Angeles CA. Expo '79 is held in conjunction with the 16th Numerical Control Society Annual Meeting and Technical Conference. Contact Numerical Control Society, 1800 Pickwick Av, Glenview IL 60025, (312) 724-7700.■



Photo 1: The IBM 704 data processing system. Along the left wall of the room is the magnetic core storage. The operator's console is in front of the central processing unit. Along the back wall is the power supply, printer, card reader and card punch. The magnetic tape units are lined up along the right side of the room.

History of Computers

The IBM 704

Keith S Reid-Green
Software Systems Development
Educational Testing Service
Princeton NJ 08540

The IBM 704, announced in 1956, was almost the last vacuum tube computer. It was rendered obsolete by the 709 in 1959, which became the 7090 in less than a year. The addition of the zero suffix to the model number denoted the replacement of vacuum tubes by transistors and the start of the second computer generation. The 704 was very definitely a first generation machine in the tradition of the *giant brains* (see photo 1) of the 1940s and 1950s. The central processing unit occupied a huge L-shaped cabinet of about 120 cubic feet. There was not enough room in the cabinet for the 32 K 36 bit words of core storage that occupied its own box — another 100 cubic feet at least. Today the computing power contained in those two steel, glass and dark gray enamel cabinets can be equaled by a desktop machine; in fact the latter would easily outstrip the 704's 12 μs instruction cycle. Nevertheless, the 704 was the ultimate computer of its time.

IBM software support for the 704 was almost nonexistent. Of course, in the days before operating systems and multiprogrammable computers, there wasn't much need for support. The assembler and a few general purpose subroutines could be acquired through SHARE (Society to Help Avoid Redundant Effort). On the other hand, support for the hardware was much in evidence. A customer engineer spent eight hours a day on the premises, including an hour long daily preventive maintenance session before the normal first shift began. The session included bringing up power, which took 15 minutes, and running diagnostics with voltages 20 percent above and below normal in an effort to find weak vacuum tubes. The effort was usually successful.

Before the advent of FORTRAN in 1958, written material for 704 programmers could be found in a single manual, *IBM Reference Manual, 704 Data Processing System.* Its

100 pages contained a description of the instruction set, operating instructions, an explanation of the virtues of assembler language coding, programming examples, explanations of the binary number system and binary arithmetic, and octal to decimal conversion tables.

The 704's input and output (IO) units consisted of a card reader, card punch, line printer, ten magnetic tape units, two magnetic drum units (each composed of four 2 K word drums) and a cathode ray tube. Off line, or peripheral, equipment consisted of three machines to process magnetic tape: card-to-tape, tape-to-card and tape-to-printer transfer devices. Important central processor features included floating point hardware and three index registers.

IBM designed the on line card reader for binary cards; cards containing machine language programs in binary object format. Since the 704 word length was 36 bits, cards were read row-wise into storage. Columns 1 thru 36 of the 9 row were read first, followed by columns 37 thru 72 of the 9 row, 1 thru 36 of the 8 row, etc. Columns 73 thru 80 could not be read; this restriction governs the present day format of the FORTRAN statement. Binary coded decimal punched cards, the kind that came out of keypunch machines, could be read by the off line card to tape transfer machine, but naturally it was frequently necessary to read such cards on line. A rather complex subroutine was required to convert these cards to internal 6 bit binary coded decimal, since they were punched column-wise and read row-wise.

The ten magnetic tape units were referred to in programming by unit numbers 0 thru 9. Unit numbers were dialed by the operator on each tape unit, hence it was possible to write two or three copies of a tape file simultaneously. Of course, the possibility of accidentally trying to read two files at once also existed, invariably resulting in a yellow *read check* light, a halted computer, and an embarrassed operator.

Tape was moved at 75 inches (190.5 cm) per second, transferring 200 bytes per inch (2.5 cm) in 7 track format. Byte and track parities were generated and checked automatically, although the words byte and parity were not yet used. Parity checking was called *redundancy checking,* and a byte was a *character.*

Probably the most remarkable piece of equipment on the 704 was the cathode ray tube. This early attempt to provide computer graphics capability was only partly successful. A 1024 by 1024 raster was addressable, but each point had to be

All photos reproduced by permission of IBM Corporation.

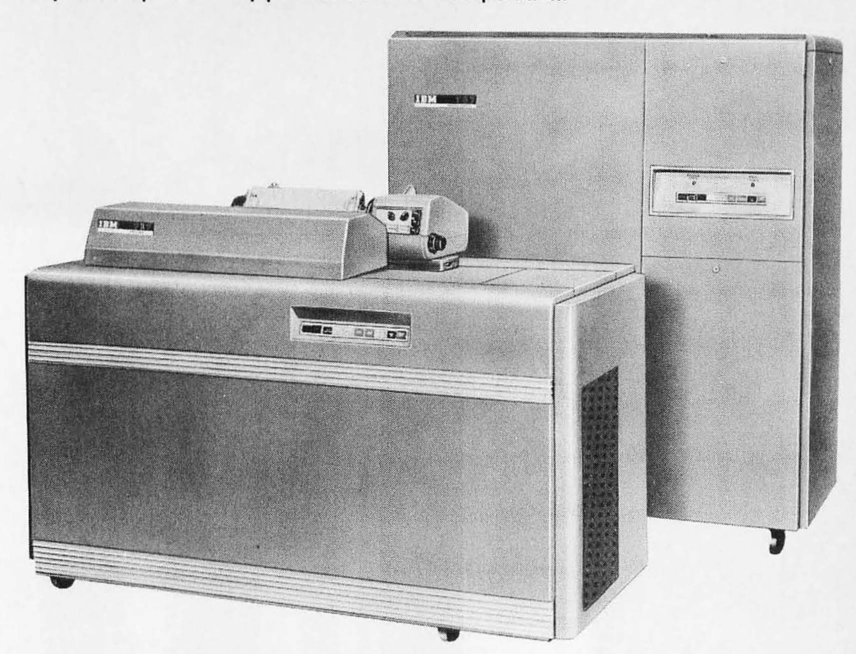


Photo 2: The off line printer was coupled to a 727 tape unit by a 757 printer control unit. This and other peripherals permitted use of magnetic tape as the 704's primary medium of input and output.

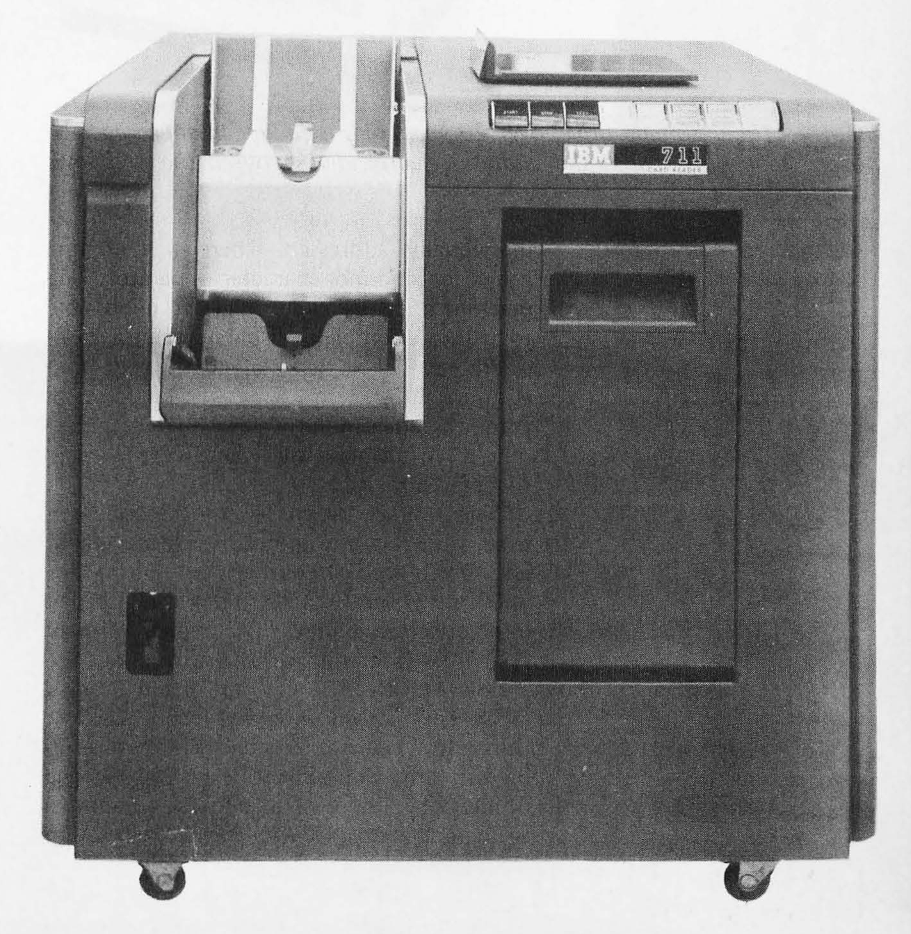


Photo 3: The 711 on line card reader read 24 36-bit numbers in row binary format from columns 1 thru 72. Columns 73 thru 80 could not be read.

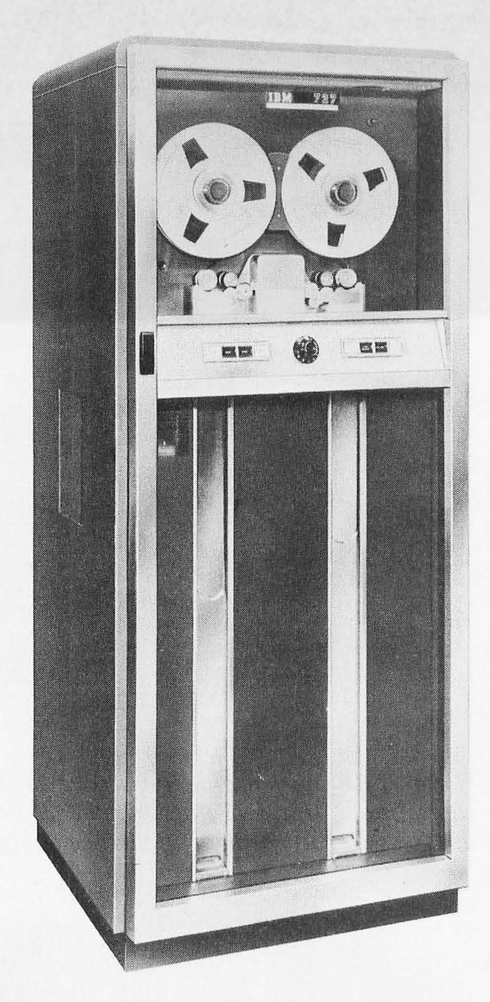


Photo 4: The 727 tape drive, over 20 years old, is similar in appearance to modern tape units. Note the address selector dial between the two rows of buttons.

individually addressed. There was no line generator and no character generator. The machine's 12 μs cycle made it impossible to run a complex display without considerable flicker. One of the big demonstrations of the time was a tic-tac-toe program. Though only a few lines were required to display the game, the display flickered noticeably.

The primary intent of the cathode ray tube was for plotting, not interactive graphics. On a dedicated machine renting for $600 per hour, who could have afforded interactive graphics? The video display system was composed of two units: the visible 21 inch (53.3 cm) screen and a 7 inch (17.8 cm) screen in a lightproof box. A shutterless camera recorded the picture on 35 mm film. Consequently, the 704 was one of the very few general purpose computers to include a *"change film frame"* instruction in its repertoire.

The central processing unit, or *calculator* as it is repeatedly called in the manual, centered around two registers: the accumulator and the multiplier-quotient (MQ) register. The accumulator performed addition, subtraction and logical operations. The MQ register served not only in multiplication and division but also as the register through which input and output transfers were made. Strangely, the accumulator was 38 bits long. It was composed of a sign and 35 numeric bits for arithmetic operations, a P bit used as the high order bit instead of the sign in overflow, shifting and logical operations, (ie: Boolean and byte manipulations), and a Q bit to the left of the P bit. The sole function of the Q bit seems to have been to generate obscure coding errors.

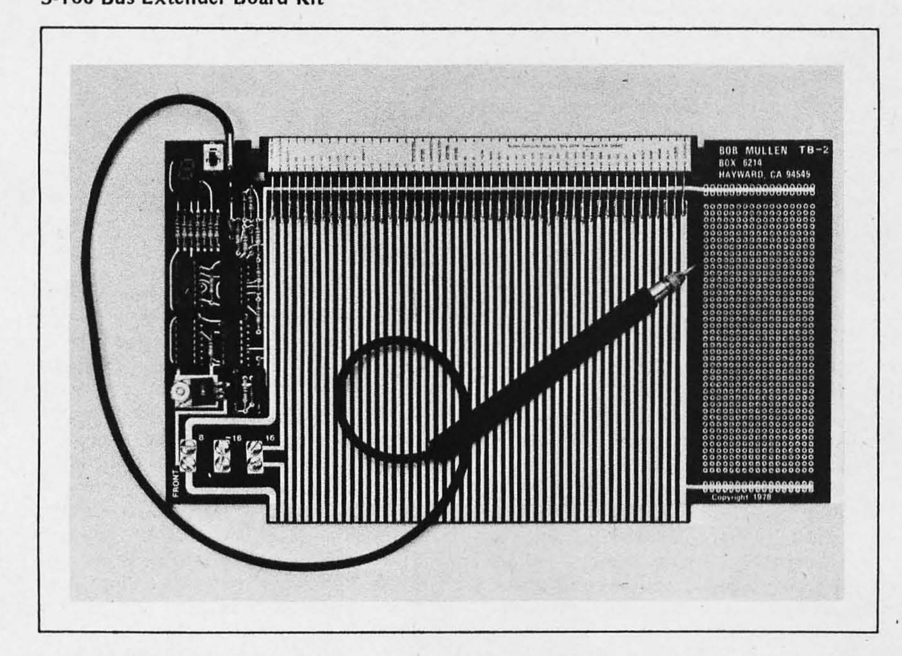
The 32,768 word maximum core storage required a 15 bit address portion in storage reference instructions. Three index registers, known to the 704's designers as registers A, B and C, were invariably referred to as indices 1, 2 and 4 by programmers, since the registers corresponded to those bits in the instruction's tag field. Indexing was subtractive, done by 2's complement arithmetic, and was another minor source of annoyance to programmers. Because of the subtractive indexing scheme, it was the usual practice to index through a table starting with the last word and moving toward the beginning.

Many other features of the 704 were a challenge to handle, but nevertheless the machine and its programmers contributed to more than just computer folklore. Several FORTRAN attributes that were developed on the 704 persist to the present day, for better or worse. The FORMAT statement, for example, was lifted bodily from an input subroutine originally meant to read punched cards and to convert specified fields to integer binary, floating point or binary coded decimal. The three branch IF statement, now mercifully replaced by a more comprehensible IF-THEN type, was designed around the 704's *compare accumulator with storage* instruction. This instruction processed the following instruction, or skipped one or two instructions depending on the accumulator being less than, equal to, or greater than the specified storage location.

Most of the numerical methods associated traditionally with Newton's iteration or infinite series expansions were developed and in many cases refined on the 704. Thus, the software subroutines that are presently used in personal computers, intelligent terminals and microprocessors owe their start to 704 programmers and applied mathematicians. The next time you take a square root or draw a circle on your video tube, remember the 704 — and be glad it's not around any more.■

What's New?

S-100 Bus Extender Board Kit



The Mullen TB-2 Extender Board kit retains the price of its predecessor and offers several new features. The built-in logic probe reads out into a 7 segment display, and includes a pulse catcher plus light emitting diode whose brightness corresponds to the duty cycle of a pulse stream. A general purpose kluge board section, with holes on 0.1 inch (0.25 cm) grid, aids development of circuits used in debugging or testing; an on board 5 V at 1 A regulator powers this section as well as the logic probe.

The TB-2 incorporates features of the previous model, such as links in the power supply lines for current measurement, fusing and independent supply switching; an edge connector label that identifies power, ground and S-100 bus signal pins; full width board size to allow use of card guides; and gold plated edge connector teeth that stand up to repeated insertion.

The Mullen Extender Board kit lists for $35. Contact Mullen Computer Products, POB 6214, Hayward CA 94545.■

Circle 541 on inquiry card.

High Level Programming Language

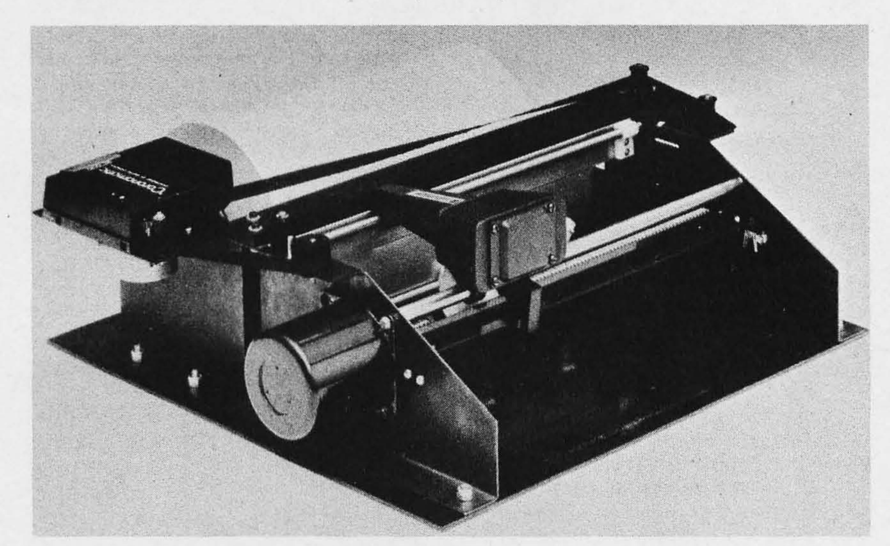
SAM76 is an interactive high level computer language that introduces concepts for both the home computer and timesharing system users. It was designed to be simple enough for the novice to learn while meeting all the requirements of the most sophisticated users.

ASCII character is the only data type, allowing data to be anything from complicated commands to syntax characters. SAM76 mathematical capability includes arbitrary precision arithmetic. Over 150 defined built-in functions give the advanced user the ability to do such things as handle arbitrary size lists and treat them as data.

Object code is available on paper tape or TDL cassette format in several configurations for the Z-80 or 8080. A PolyMorphic version on cassette tape and a CP/M disk version are available. All versions are coded for Apple and Zapple conventions with patch information on input and output (IO) vectors available. The SAM76 200 page language manual is available for $12. The object code is $6 (CP/M is $10 for disk) and $2 for the IO information. For further information contact SAM76, POB 257, Pennington NJ 08534.■

Circle 542 on inquiry card.

Printer Mechanism for the Personal Computer Experimenter

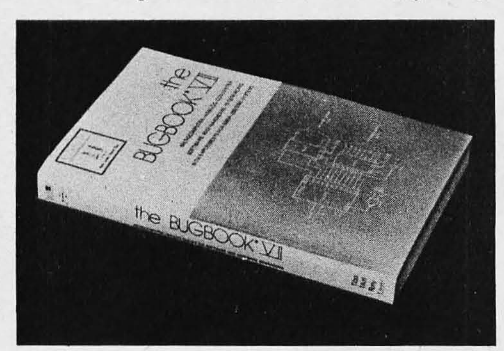


This printer mechanism is for the personal computer user who is interested in constructing a bidirectional dot matrix impact printer. Print speed is 120 characters per second with a line capacity of 80 characters. It uses standard 8.5 inch (21.25 cm) single or multiple copy paper. The paper is advanced by motor and pressure roller with a slew rate of 400 lines per minute. The mechanism uses ribbon cartridge with an independent motor. The unit has optoisolators for beginning of line, end of line and character positioning sensing. The price is $399 including electronic interface. Write to MarComm Inc, POB 535, Ramona CA 92065.■

Circle 543 on inquiry card.

PUBLICATIONS

Bugbook VII Teaches Microcomputer-Analog Converter Interfacing



The Bugbook VII, Microcomputer-Analog Converter Software and Hardware Interfacing augments the Bugbook series with an in-depth treatment of how 8080 family microcomputers are interfaced to real world analog devices for measurement, control and display applications. The 284 page soft-cover book combines practical examples of hardware and software analog converter interfacing techniques with a series of experiments in waveform generation, data acquisition and video display control.

Bugbook VII begins with digital to analog converter interfacing, and proceeds to software control of ramp, successive approximation, and dual slope analog to digital converters. Data acquisition approaches using software interrupts and real time clocks are compared. Sample and hold amplifiers and analog multiplexers used in many practical data acquisition systems are treated in a full chapter. Consideration is also given to the factors in selecting and interfacing packaged analog to digital and digital to analog data acquisition modules.

The book is priced at $8.50 and is available from E&L Instruments Inc, 61 First St, Derby CT 06418.■

Circle 611 on inquiry card.

Texas Instruments Offers Free Linear and Interface Circuits Guide



A new *Linear and Interface Circuits Master Selection Guide* (CL-329) is available free from Texas Instruments Inc. This 128 page product selection guide and catalog is designed to provide a reference to Texas Instruments' linear and interface circuits.

This publication covers BIFET and bipolar operational amplifiers, voltage regulators, voltage comparators, line circuits, peripheral, MOS, memory and display drivers, sense amplifiers and special functions. Special function monolithic integrated circuits such as analog switches, analog to digital converters for processors, precision timers, and Hall effect switches are described with schematics, packaging information and key features. Basic descriptions and features are presented in short form to help users to select the proper integrated circuit.

The CL-329 Master Selection Guide contains a complete cross-reference guide which includes package and temperature designations. Write to Texas Instruments Inc, Inquiry Answering Service, POB 5012, M/S 308 (Attn: CL-329), Dallas TX 75222.■

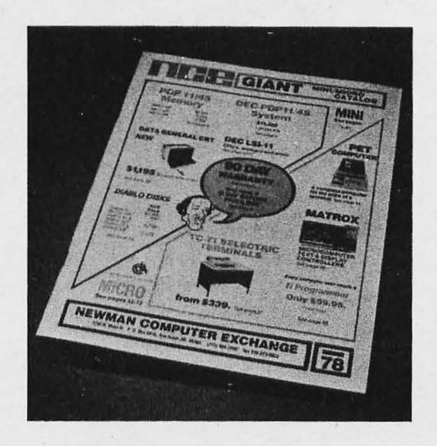
Circle 613 on inquiry card.

Newman Computer Exchange Catalog Available

This 72 page minicomputer and microcomputer catalog is available from Newman Computer Exchange, 1250 N Main St, Ann Arbor MI 48104. The catalog lists a wide selection of new and used minicomputer and microcomputer products.

The minicomputer section lists primarily Digital Equipment and Data General items, while the microcomputer products include a variety of personal computers along with books, accessories, and hundreds of other items of interest to the computer enthusiast.■

Circle 614 on inquiry card.



New Book on Home Computer Applications



The Mind Appliance: Home Computer Applications by T G Lewis is written for the serious personal computer user who wants to utilize the computer as a household appliance. This book contains dozens of BASIC language programs which will enable your computer to balance a checkbook, automatically dial the telephone, handle household budgets, and plan menus. These and many other ideas, such as writing poetry, drawing figures, and scoring music, are fully illustrated by actual programs that can be put to use. The book is written in an entertaining style that leads the reader into relatively complex programming situations with a minimum of effort. This 144 page book is priced at $6.95 and is published by Hayden Book Company Inc, Rochelle Park NJ 07662.■

Circle 612 on inquiry card.

Attention Readers, and Vendors...

Where Do New Product Items Come From?

The information printed in the new products pages of BYTE is obtained from "new product" or "press release" copy sent by the promoters of new products. If in our judgment the neat new whizbang gizmo or save the world software package is of interest to the personal computing experimenters and homebrewers who read BYTE, we print the information in some form. We openly solicit such information from manufacturers and suppliers to this marketplace. The information is printed more or less as a first in first out queue, subject to occasional priority modifications.

JADE Computer Products

DISK DRIVES

B51 - 5¼" $295.00
by Micro Peripherals, Inc. Operates in either single density (125KB, unformatted) or double density (250KB, unformatted) modes, up to 40 tracks, with a track-to-track access time of only 5 ms.

SA801R $495.00
by Shugart. Single-sided 8" floppy disk drive.

8" DISK DRIVE $395.00
by GSI/Siemans. Direct equivalent of Shugart 801R.

DM 2700-S $750.00
Includes SA801R, 10"x10"x 16" cabinet, power supply, data cable, fan, AC line filter.

VERBATIM™ FLOPPY DISKS

5¼" MINIDISKETTES
Soft sector, 10-sector, or 16 sector
$4.40 each
box of 10 $40.00

8" FLOPPY DISKS
Soft sector $4.75 ea.
box of 10 $42.50

FLOPPY DISK INTERFACE

JADE FLOPPY DISK (Tarbell board)
Kit $175.00
Assm. & Tested $250.00

S.D. Computer Products "Versa Floppy"
Kit $159.95
Assm. & Tested $189.95

MINISCOPES



- 15 megahertz bandwidth.
- External and internal trigger.
- Time Base - 0.1 microseconds to 0.5 Sec/div - 21 settings.
- Battery or line operation.
- Automatic and line sync modes.
- Power consumption less than 15W
- Verticle Gain - 0.01 to 50 volts/div-12 settings.
- Weight is only 3 pounds.

MS-15 $ 318
MS-215 Dual Trace Version $435
With Rechargeable Batteries & Charger Unit
10 to 1, 10 meg probe $27.
Leather carrying case $45.

LOGIC PROBES

CSC

MODEL LP-1
Hand-held logic probe provides instant reading of logic levels for TTL, DTL, HTL or CMOS. **Input Impedance:** 100,000 ohms. **Minimum Detectable Pulse:** 50 ns. **Maximum Input Signal (Frequency):** 10 MHz. **Pulse Detector (LED):** High speed train or single event. **Pulse Memory:** Pulse or level transition detected and stored.
CSC Model LP-1 Logic Probe Net Each **$44.95**

MODEL LP-3
High speed logic probe. Captures pulses as short as 10 ns. **Input Impedance:** 500,000 ohms. **Minimum Detectable Pulse:** 10 ns. **Maximum Input Signal (Frequency):** 50 MHz. **Pulse Detector (LED):** High speed train or single event. **Pulse Memory:** Pulse or level transition detected and stored.
CSC Model LP-3 Logic Probe Net Each **$69.95**

3-LEVEL GOLD WIRE WRAP SOCKETS

14 PIN 39¢ each
16 PIN 43¢ each
100 for $30.00
Sockets are end and side stackable, closed entry.

METERS

Rechargeable batteries and charger
Measures DC Volts, AC Volts, Ohms and Current
Automatic polarity, decimal and overload indication
No zero adjustment and no full-scale ohms adjust
Battery-operated — NiCad batteries; also AC line operation.
Large LED display for easy reading without interpolation
Size: 1.9"H x 2.7"W x 4"D
Parts & labor guaranteed 1 year
Tilt stand option $3.50
Leather case $20.00

LM3A 3 dig 1% DC $134.00
LM3.5A 3½ dig .5% DC $158.50
LM40A 4 dig .1% DC $209.00
LM4A 4 dig .03% DC $250.00

S-100 MOTHER BOARDS

JADE 6-SLOT
Kit $41.95
Assembled & Tested $56.95
Bare Board $24.95

9-SLOT "LITTLE MOTHER"
Kit $85.00
Assembled & Tested $99.00
Bare Board $35.00

13-SLOT "QUIET MOTHER"
Kit $95.00
Assm. & Tested $110.00
Bare Board $40.00

22-SLOT "STREAKER"
Assm. & Tested $149.00

GOLD PLATED S-100 EDGE CONNECTORS

Soldertail $3.25 each
10 for $30.00
Wire Wrap $4.50 each
10 for $40.00

JADE VIDEO INTERFACE

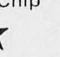
S-100 Compatible Serial Interface with Sockets Included.
Kit $117.95
Assm. & Tested $159.95
Bare Board with manual $35.00

TU-1

Convert your T.V. set into a Video Monitor
Kit $8.95

Z80A SPECIAL

4MHz Zilog CPU Chip
☆ $14.95 ☆



ELECTRONIC SYSTEMS

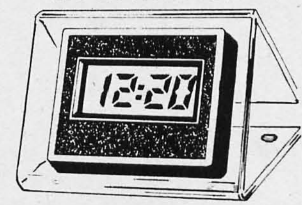
"KANSAS CITY STANDARD" TAPE INTERFACE
Part no. 111
Board $7.60 with parts $27.50

RS-232/TTL INTERFACE
Part no. 232
Converts TTL to RS-232 and RS-232 to TTL
Board only $4.50; with parts $7.00

RS-232/TTY INTERFACE
Part no. 600
Converts RS-232 to 20mA current loop, and 20mA current loop to RS-232
Board only $4.50, with parts $7.00

UART & BAUD RATE GENERATOR
Part no. 101
Converts serial to parallel and parallel to serial
Board only $12.00; with parts $35.00
with connector add $4.00

LIQUID CRYSTAL DIGITAL CLOCK-CALENDAR



- For Auto, Home, Office
- Small in size (2x2½x½)
- Push button for seconds release for date.
- Clocks mount anywhere with either 3M double-sided tape or VELCRO, included.
- 2 MODELS AVAILABLE:
  LCD-101, portable model runs on self-contained batteries for better than a year.
  LCD-102, runs on 12 Volt system and is back lighted.
- LCD-101 or LCD-102 your choice **$34.95..**
- Clear desk stand for **$2.00**

TARBELL CASSETTE INTERFACE

S-100 cassette interface with speeds from 187 to 540 bytes per second.
JADE Kit $99.95
Assembled $135.00

JADE PARALLEL/SERIAL INTERFACE

S-100 compatible, 2 serial I/O ports, 1 parallel I/O.
Kit JG-P/S $124.95
Ass. & Tested JG-P/SA $179.95
Bare Board with manual $30.00

See opposite page for ordering instructions.

VECTOR SLIT-N-WRAP™

WIRE WRAPPING TOOLS for 0.025 inch (.64 mm) SQUARE POSTS

Slit-N-Wrap wiring is easy to use and is up to four times faster than ordinary manual wrapping because there is no wire measuring or pre-stripping required. Slit-N-Wrap tools have a patented action which slits wire insulation while the tool is wrapping wire on 0.025 inch (.64 mm) square posts. Connections are just as reliable as with conventional wrapping tools. Tests show Slit-N-Wrapped connections exceed MIL Spec requirements for pull-off, are low resistance (less than .003 ohms) and are gas tight. All tools and bits are guaranteed to provide at least 10,000 reliable wraps before bit replacement is required.

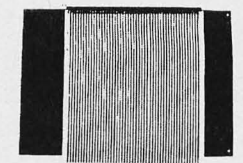
P180 Series Tools Using Polyurethane Nylon Coated Wire.
Polyurethane nylon coated copper wire is used in all P180 series tools. The small diameter of this insulated 28 gage wire permits two 7-turn wraps on a 0.025 inch (.64 mm) square post occupying only .21 inch (5.3 mm) of post length. Soldering is not required on rectangular posts, but if wrapped on round or irregular posts it may be soldered using a 750°F (399°C) iron which melts the insulation as solder flow occurs.

View of unwrapped wire P180 series wired post

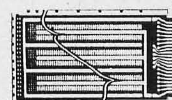
"daisy-chain" wraps

P180 Manual Wrapping Tool $24.50
P180-4T NiCad Battery powered P180 $75.00
P180-4T1 AC powered version of P180 $80.00
All P180 Series Tools include two 100 ft. spools of polyurethane nylon insulated wire, P183 forming knife and chisel tool, complete instructions.

WIRE
Polyurethane nylon insulated copper 28 gage wire come on 100 ft. spools as follows:
W28-2A pkg. 3 GREEN $2.75 **W28-2C** pkg. 3 CLEAR $2.75
W28-2B pkg. 3 RED $2.75 **W28-2D** pkg. 3 BLUE $2.75



3690-12 CARD EXTENDER
Card Extender has 100 contacts-50 per side on .125 centers-Attached connector is compatible with S-100 Bus Systems $25.00.
3690 6.5" 22/4 pin. 158 ctrs. Extenders $12.00



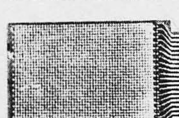
Gen. Purpose D.I.P. Boards with Bus Pattern for Solder or Wire Wrap. Epoxy Glass 1/16" 44 pin con. spaced .156
3677 9.6" x 4.5" $10.90
3677-2 6.5" x 4.5" $9.74

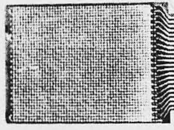
Vector Plugboards

8800V
Universal Microcomputer/processor plugboard use with S-100 bus. Complete with heat sink & hardware 5.3 x 10 x 1/16

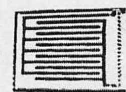
8801-1
Same as 8800V except plain, less power buses & heat sink

	1-4	5-9	10-24
8800V	19.95	17.95	15.96
8801-1	14.95	13.46	11.96





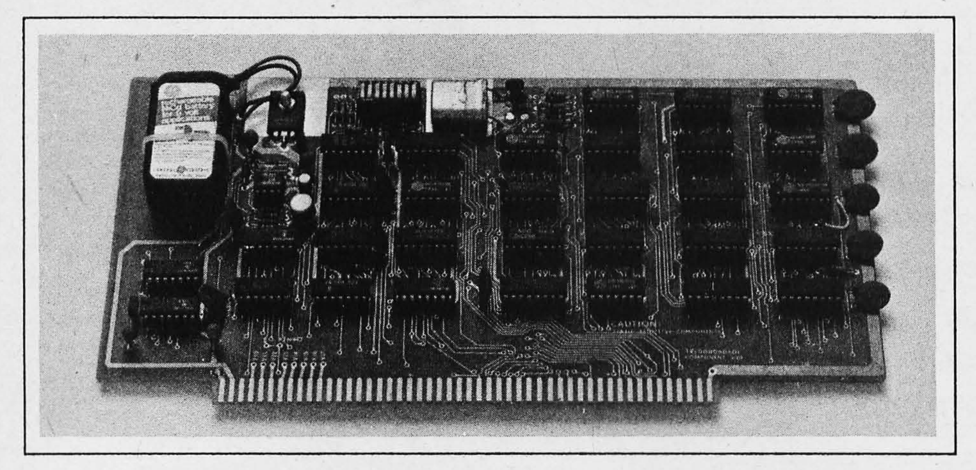
P pattern plugboards for IC's Epoxy Glass 1/16" 44 pin con. spaced .156
3662 6.5" x 4.5" $7.65
3662-2 9.6" x 4.5" $11.45



Hi-Density Dual-In-Line Plugboard for Wire Wrap with Power & Grd. Bus Epoxy Glass 1/16" 44 pin con. spaced .156
3682 9.6" x 4.5" $10.97
3682-2 6.5" x 4.5" $9.81

Circle 195 on inquiry card.

New Clock Board Features Battery Backup and Crystal Control



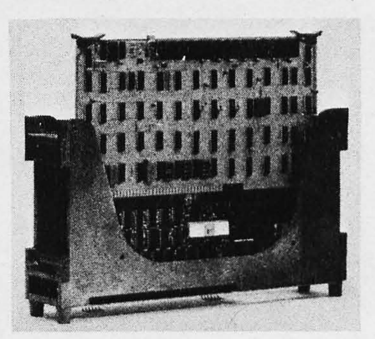
This board keeps time in 100 μs increments for periods as long as 100,000 days. An interrupt feature has been provided which can be programmed for any change in a clock digit to help make efficient use of computer time. The clock is crystal controlled for accuracy and an on board, 9 V rechargeable battery keeps the clock running during computer down times.

It is set by entering binary coded decimal digits at each time port. The clock stops the moment the first digit is entered and starts again on the first "read" command. A "write protect" switch prevents the clock from being accidentally stopped or changed. The clock can be used with most BASICs.

The price of the 100,000 day clock is $219 assembled and tested, and $179 in kit form. Write to Mountain Hardware Inc, 5523A Scotts Valley Dr, Scotts Valley CA 95066.■

Circle 577 on inquiry card.

Video Boards Interface Video Monitor to Intel Single Board Computer Bus



The MSBC family of video interface boards are designed to interface a video monitor to the Intel single board computer bus. The two cards in the family may be used either singly or in combination to provide a variety of alphanumeric and graphic display capabilities. Graphic resolution ranging from 256 by 256 to 512 by 512 points may be obtained from the MSBC-512, while a 24 line by 80 character display is produced by the MSBC-2480. Multiple cards provide limitless possibilities for color or grey scale imaging. A set of software packages are available to facilitate use of the single board computer system. For further information about the MSBC family, contact Matrox Electronic Systems, POB 56, Ahuntsic Station, Montreal, Quebec CANADA H3L 3N5.■

Circle 578 on inquiry card.

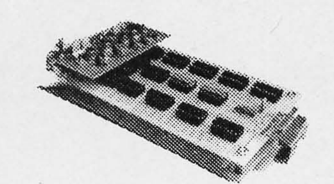
VISIBLE OR INFRA RED	USED FOR CHARACTER RECOGNITION FOR COMPUTERS WITH EXTERNAL CIRCUITS	MAY BE USED IN A VACUUM, UNDER WATER, HIGH ALTITUDE	IN MAGNETIC ENVIRONMENT BECAUSE THERE IS NO HIGH VOLTAGE OR MAGNETIC DEFLECTION

MINATURE SOLID STATE

202 VIDEO CAMERA KIT

FEATURING A. . . 100 × 100 BIT SELF SCANNING CHARGED COUPLED DEVICE

THIS UNIQUE UPDATED CAMERA KIT
FEATURES THE FAIRCHILD CCD 202C IMAGE SENSOR

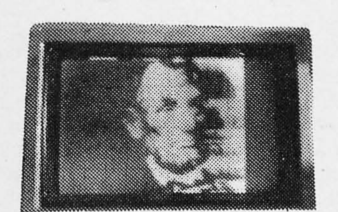


ADVANTAGES

- **IN THE FUTURE WE WILL SUPPLY A COMPUTER VIDEO INTERFACE CARD**
- All clock voltages operate at 6V reguiring no adjustments
- Higher video output signal
- We supply the power board, so only a 5V 1 Amp power source is needed
- The circuitry has been simplified for easier assembly
- Two level TTL output is supplied for interfacing

FEATURES

- Sensitive to infra red as well as visible light
- May be used for IR surveillance with an IR light source
- Excellent for standard surveillance work, because of light weight and small size
- All components mounted on parallel 3¾" × 6½" single sided boards
- Total weight under 1 lb.



We supply all semiconductors, boards, data sheets, diagrams, resistors and capacitors, and 8MM lens.
Sorry we do not supply the case, batteries and 5V supply.

$349.00 KIT

Add $75.00 to assemble and test

Add $2.00 Postage and Handling

UNIVERSAL 4Kx8 MEMORY BOARD KIT

$69.95

32-2102-1 fully buffered, 16 address lines, on board decoding for any 4 of 64 pages, standard 44 pin buss, may be used with F-8 & KIM

EXPANDABLE F8 CPU BOARD KIT

$99.00

featuring Fairbug PSU.1K-of static ram, RS 232 interface, documentation, 64 BYTE reqis.er

C/MOS (DIODE CLAMPED)

4001 –	.18	4019 –	.37	4049 –	.35	74C73 –	.65
4002 –	.18	4020 –	.90	4050 –	.35	74C74 –	.45
4006 –	.95	4021 –	.90	4053 –	1.10	74C83 –	1.15
4007 –	.18	4022 –	.90	4055 –	1.25	74C86 –	.40
4009 –	.37	4023 –	.18	4066 –	.70	74C93 –	.75
4010 –	.37	4024 –	.75	4071 –	.18	74C151	1.40
4011 –	.18	4025 –	.18	4076 –	.97	74C160	1.05
4012 –	.18	4027 –	.37	4520 –	.70	74C161	1.05
4013 –	.29	4028 –	.80	74C00 –	.22	74C174	1.05
4014 –	.75	4029 –	.95	74C02 –	.22	74C175	1.05
4015 –	.75	4030 –	.33	74C04 –	.24	74C193	1.20
4016 –	.29	4035 –	.97	74C08 –	.22	74C901	.48
4017 –	.90	4042 –	.65	74C10 –	.27	74C902	.48
4018 –	.90	4046 –	1.35	74C42 –	.85	74C914	1.70

1488 RS232 INTERFACE	$ 1.40	2114 4K STATIC RAM (450 ns)	$ 8.50
1489 RS232 INTERFACE	$ 1.25	4116	$ 15.95
2509 TRI-STATE STATIC SR	$ 1.35	5280/2107B 4K DYN RAM	$ 3.40
2708 8K EPROM (450 ns)	$ 8.75	TMS 4050L	$ 3.95
TMS 3409 NC 80 BITS DYN. S.R.	$1.35	5204 4K PROM	$ 4.95
2716 (TI)	$24.95	82S23	$ 1.95
2522 STATIC SHIFT REG	$ 1.95	82S129	$ 3.25
2513 CHARACTER GEN	$ 6.75	AY 5 1013 UART	$ 5.25
2518 HEX 32 BIT SR	$ 2.20	TR1602B	$ 4.95
2102 1 (450 ns)	$ .99	8703C TELEDYNE 8 BIT A/D CONVERT	$ 9.95
21L02 1 (450 ns)	$ 1.25	8080A	$ 7.75
MM5270 4K XI DYN	$ 3.45	8224	$ 3.95
MK 4008P	$ 1.95	8228	$ 4.95
2101 1 254 x 4 STATIC	$ 2.45	AMD 9214DC 512x8 ROM (450NS)	$ 5.75
2111 1 256 x 4 STATIC	$ 3.45		
2112 1 256 x 4 STATIC	$ 2.75		

CRYSTALS $3.45 ea.
4.000 MHz
5.000 MHz
6.000 MHz
8.000 MHz
10.000 MHz

RIBBON CABLE FLAT (COLOR CODED) #30 WIRE
26 cond. - .50/per foot
40 cond. - .75/per foot
50 cond. - .90/per foot

CTS 206-8 eight position dip switch $1.60
CTS-206-4 four position dip switch $1.45
LIGHT ACTIVATED SCR's TO-18, 200V 1A. .... $ .70

SILICON SOLAR CELLS
2¼" diameter 4V at 500 ma $4.00

LED READOUTS

FND 359 C.C. .4"	$ .50	DL-704 C.A. .3"	$ .75
FCS 8024 4 digit C.C. 8" display	$5.95	DL 747 C.A. .6"	$1.95
FND 503 C.C. .5"	$ .85	FND 803 C.C. .8"	$1.95
FND 510 C.A. .5"	$ .85	FND 810 C.A. .8"	$1.95
DL-704 .3" C.C.	$ .85		

PRINTED CIRCUIT BOARD

4-1 2"x6-1 2" SINGLE SIDED EPOXY BOARD 1 16" thick, unetched
$.60 ea. .... 5/$2.60

7 WATT LD-65 LASER DIODE IR $8.95

2N 3820 P FET .... $ .45
2N 5457 N FET .... $ .45
2N2646 UJT .... $ .45
ER 900 TRIGGER DIODES 4 $1.00
2N 6028 PROG. UJT .... $ .65

MINIATURE MULTI-TURN TRIM POTS
100, 1K, 2K, 5K, 10K, 20K, 50K, 100K, 200K, 500K, 1Meg, 2Meg, $.75 each 3/$2.00

CHARGED COUPLE DEVICES

CCD 201C 100x100 Image Sensor $95.00
CCD 202C 100x100 Image Sensor $145.00

VERIPAX PC BOARD.... $4.00
This board is a 1/16" single sided paper epoxy board, 4½"x6½" DRILLED and ETCHED which will hold up to 21 single 14 pin IC's or 8,16 or LSI DIP IC's with busses for power supply connector.

FP 100 PHOTO TRANS .... $ .50
RED, YELLOW, GREEN or AMBER LARGE LED's .2" .... 6/$1.00
TIL-118 OPTO-ISOLATOR .... $ .75
MOLEX PINS .... 100/$1.00
1000/$8.00

1 WATT ZENERS: 3.3, 4.7, 5.6, 9.1, 10, 12, 15, 18, or 22V .... 6/$1.00
MC6860 MODEM CHIP $9.95
MCM 6571A 7 x 9 character gen ... $10.75

Silicon Power Rectifiers

PRV	1A	3A	12A	50A	125A	240A
100	.06	.14	.30	.80	3.70	5.00
200	.07	.20	.35	1.15	4.25	6.50
400	.09	.25	.50	1.40	6.50	9.50
600	.11	.30	.70	1.80	8.50	12.50
800	.15	.35	.90	2.30	10.50	16.50
1000	.20	.45	1.10	2.75	12.50	20.00

SAD 1024-a REDICON 1024 stage analog "Bucket Brigade" shift register. $14.95

IN 4148 (IN914) .... 15/$1.00

RS232 CONNECTORS
DB 25P male $2.25
DB 25S female $2.95
HOODS $1.00

REGULATORS

309K	$ 1.20	340K-12,15 or 24V	$ .95
723	$ .50	340T-5, 6, 8, 12 15,18 or 24V	$ .95
LM 376	$ .60	78 MG	$1.35
320T-12, 15 or 24V	$1.25	79 MG	$1.35

TRANSISTOR SPECIALS

2N6233-NPN SWITCHING POWER $ 1.95
MRF-8004 a CB RF Transistor NPN $ .75
2N3772 NPN Si TO-3 .... $ 1.00
2N1546 PNP GE TO-3 .... $ .75
2N4908 PNP Si TO-3 .... $ 1.00
2N6056 NPN Si TO-3 Darlington $ 1.70
2N5086 PNP Si TO-92 .... 4 $ 1.00
2N3137 NPN Si RF .... $ .55
2N3919 NPN Si TO-3 RF .... $ 1.50
2N1420 NPN Si TO 5 .... 3/$ 1.00
2N3767 NPN Si TO-66 .... $ .70
2N2222 NPN Si TO-18 .... 5/$ 1.00
2N3055 NPN Si TO-3 .... $ .50
2N3904 NPN Si TO-92 .... 6/$ 1.00
2N3906 PNP Si TO-92 .... 6/$ 1.00
2N5296 NPN Si TO-220 .... $ .50
2N6109 PNP Si TO-220 .... $ .55
2N3638 PNP Si TO-5 .... 5/$ 1.00
MPSA 13 NPN Si .... 4/$ 1.00

TTL IC SERIES

				74151–	.61
		7445–	.65	74153–	.61
7400–	.13	7446–	.68	74154–	.94
7401–	.13	7447–	.58	74155–	.58
7402–	.13	7448–	.68	74157–	.55
7403–	.13	7450–	.15	74161–	.55
7404–	.15	7472–	.25	74162–	.79
7405–	.13	7473–	.28	74163–	.55
7406–	.16	7474–	.28	74164–	.85
7407–	.20	7475–	.45	74165–	.95
7408–	.18	7476–	.30	74170–	1.68
7409–	.18	7480–	.31	74173–	1.20
7410–	.13	7483–	.65	74174–	.95
7411–	.18	7485–	.87	74175–	.85
7412–	.13	7486–	.28	74176–	.75
7413–	.36	7489–	1.25	74177–	.75
7414–	.60	7490–	.42	74180–	.65
7416–	.22	7491–	.58	74181–	1.90
7417–	.25	7492–	.43	74190–	1.00
7420–	.13	7493–	.43	74191–	1.00
7425–	.25	7494–	.67	74192–	.79
7426–	.22	7495–	.65	74193–	.79
7427–	.19	7496–	.65	74194–	.80
7430–	.13	74107–	.28	74195–	.50
7432–	.22	74121–	.29	74196–	.86
7437–	.21	74122–	.38	74279–	.55
7438–	.21	74123–	.45	74367–	.50
7440–	.13	74125–	.40	75325–	1.50
7441–	.70	74126–	.40	75491–	.50
7442–	.37	74150–	.94	75492–	.50

DATA CASSETTES 1/2 HR $ .95

22/44 Pin Solder Tail .156" Conn. $1.95

MM 5387AA new clock chip which will directly drive LED's 12/24 hrs., 1 supply & alarm $5.95

NO. 30 WIRE WRAP WIRE SINGLE STRAND 100' $1.40

ALCO MINIATURE TOGGLE SWITCHES
MTA 106 SPDT .... $ .95
MTA 206 DPDT .... $ 1.70
MTA 206 P-DPDT CENTER OFF $ 1.85
MSD 206 P-DPDT CENTER OFF LEVER SWITCH .... $ 1.85

Full Wave Bridges

PRV	2A	6A	25A
100			1.30
200	.75	1.25	2.00
400	.95	1.50	3.00
600	1.20	1.75	4.00

DIP SOCKETS

8 PIN .17 24 PIN .35
14 PIN .20 28 PIN .40
16 PIN .22 40 PIN .60
18 PIN .25

SANKEN AUDIO POWER AMPS

Si 1010 G 10 WATTS .... $ 7.80
Si 1020 G 20 WATTS $15.70
Si 1050 G 50 WATTS $28.50

TANTULUM CAPACITORS

.22UF 35V 5/$1.00	6.8UF 35V 4/$1.00
.47UF 35V 5/$1.00	10UF 10V $ .25
.68UF 35V 5/$1.00	22UF 25V $ .40
1UF 35V 5/$1.00	15UF 35V 3/$1.00
2.2 UF 20V 5/$1.00	30UF 6V 5/$1.00
3.3UF 20V 4/$1.00	47UF 20V $ .35
4.7UF 15V 5/$1.00	68 UF 15V $ .50
	100 UF 10V $ .40

74LS SERIES

74LS00 –	.19	74LS132 –	.70
74LS02 –	.19	74LS138 –	.70
74LS03 –	.19	74LS139 –	.70
74LS04 –	.21	74LS151 –	.59
74LS05 –	.21	74LS153 –	.59
74LS08 –	.21	74LS155 –	.62
74LS09 –	.21	74LS156 –	.62
74LS10 –	.20	74LS157 –	1.59
74LS11 –	.20	74LS160 –	.79
74LS13 –	.40	74LS161 –	.79
74LS14 –	.80	74LS162 –	.79
74LS15 –	.21	74LS163 –	.79
74LS20 –	.21	74LS168 –	.80
74LS21 –	.21	74LS169 –	.80
74LS22 –	.21	74LS170 –	1.40
74LS26 –	.29	74LS173 –	.95
74LS27 –	.25	74LS174 –	.72
74LS28 –	.26	74LS181 –	2.00
74LS30 –	.21	74LS190 –	.88
74LS32 –	.25	74LS191 –	.88
74LS37 –	.28	74LS192 –	.88
74LS38 –	.28	74LS193 –	.88
74LS40 –	.21	74LS195 –	.50
74LS47 –	.73	74LS196 –	.75
74LS51 –	.25	74LS197 –	.75
74LS54 –	.25	74LS257 –	.68
74LS73 –	.35	74LS258 –	.68
74LS74 –	.34	74LS265 –	.25
74LS75 –	.50	74LS279 –	.50
74LS76 –	.35	74LS290 –	.60
74LS85 –	.75	74LS293 –	1.52
74LS86 –	.35	74LS365 –	1.49
74LS90 –	.50	74LS366 –	.49
74LS92 –	.50	74LS367 –	.49
74LS93 –	.50	74LS368 –	.49
74LS109 –	.35	74LS375 –	.60
74LS112 –	.35	74LS386 –	.35
74LS113 –	.35	74LS390 –	1.45
74LS114 –	.35	74LS670 –	2.00
74LS125 –	.45		
74LS126 –	.45		

LINEAR CIRCUITS

LM 101	– .75
LM 301/748	.25
LM307	– .30
LM 308	– .75
LM 311	– .75
LM 318	– 1.20
LM 324	– .70
LM 339	– 1.10
LM 358	– .70
LM 370	– 1.15
LM 377	– 1.60
LM 380	– .95
LM 381	– 1.25
LM 382	– 1.25
LM 387	– 1.25
LM 537	– 2.50
LM 553	– 2.50
LM 555	– .39
LM 556	– .85
NE540L	– 2.25
560	– 2.00
565	– .95
566	– 1.10
567	– 1.10
703	– .90
733H	– .75
592H	– .95
709	– .25
710	– .35
741C or V	– .25
747	– .50
LM 1310	– 2.50
1456	– .95
1458	– .50
3900	– .40
8038CC	– 3.90
791	– 1.95
LF356H	– 1.20

TRIACS / SCR'S

PRV	TRIACS 1A	TRIACS 10A	TRIACS 25A	SCR'S 1.5A	SCR'S 6A	SCR'S 35A
100	.40	.70	1.30	.40	.50	1.20
200	.70	1.10	1.75	.60	.70	1.60
400	1.10	1.60	2.60	1.00	1.20	2.20
600	1.70	2.30	3.60		1.50	3.00

Terms: FOB Cambridge, Mass. Send Check or Money Order. Include Postage, Minimum Order $5.00, COD'S $20.00

Send 25¢ for our catalog featuring Transistors and Rectifiers
145 Hampshire St., Cambridge, Mass.



SOLID STATE SALES
P.O. BOX 74B
SOMERVILLE, MASS. 02143 TEL. (617) 547-7053

WE SHIP OVER 95% OF OUR ORDERS THE DAY WE RECEIVE THEM

Circle 340 on inquiry card.

What's New?

PERIPHERALS

Video Display Terminal Features Memory Lock



The Micro Bee 2 is an 8085A processor controlled buffered video display terminal. Among the Micro Bee 2 features is the ability to invoke the memory lock that allows the operator or host computer to lock a position of the display while entering or receiving data in the unlocked portion of the display memory. The invisible memory address pointer can be used to read to and write from the display memory, independent of visible screen functions. Standard visual attributes include normal, reverse, blink, underline, and half intensity video levels. Logical attributes include protected data fields and numeric only fields, as well as modified data field transmission.

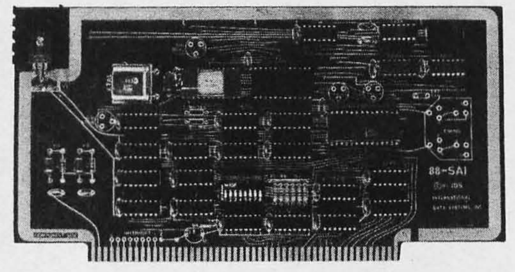
The display format is based on 24 lines of data with 96 characters in each line. 80 of the character positions are displayable, which leaves 16 nondisplayable character cells available for field attributes. The line drawing graphics capability allows for the creation of forms on the display using the vertical and horizontal line features.

The expanded characteristics of the Micro Bee 2 include a bidirectional buffered serial auxiliary port, XY addressing, read cursor address, read terminal status, time of day clock, and 128 ASCII characters with descenders on lower case characters.

The Micro Bee 2 sells for $1695 and is available from Beehive International, 4910 Amelia Earhart Dr, POB 25668, Salt Lake City UT 84125.■

Circle 615 on inquiry card.

Synchronous Interface Module from International Data Systems



The 88-SAI provides a synchronous or asynchronous interface for any S-100 bus processor. The 88-SAI is intended for use in special communications requirements such as synchronous communications between S-100 computers and large scale computers, high speed modems, data encryption devices, or other S-100 computers.

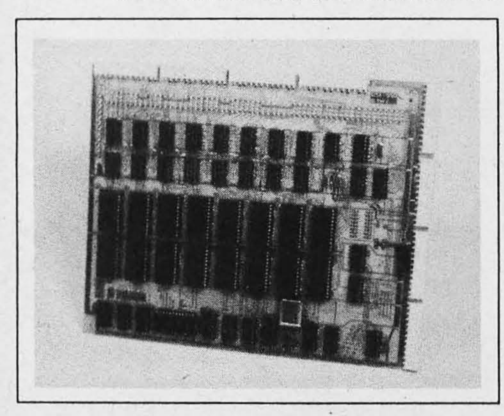
The module allows data transfer rate, word size, parity and a number of stop bits to be selected under software control. Also under software control are synchronous and asynchronous mode selection and functions associated with synchronous communications, such as the number of sync characters.

It is fully compatible with RS-232C interfaces. Additional provisions are made for interface to nonstandard devices requiring that various signals or handshake lines be inverted.

The 88-SAI is available in kit form for $199, or assembled, tested and with a limited warranty for $299. Contact International Data Systems Inc, 400 N Washington St, Suite 200, Falls Church VA 22046.■

Circle 616 on inquiry card.

16 Port Serial Board from Ohio Scientific



A new 16 port serial IO board has been announced by Ohio Scientific, 1333 S Chillicothe Rd, Aurora OH 44202. The board can be used on any Ohio Scientific computer system. It is available fully assembled as the Model CA10-X, where X specifies the number of serial ports on the board, from 2 to 16. The board features RS-232 and high speed synchronous interfaces that can be mixed in any combination. The communications transfer rate of each serial port is jumper selectable from a crystal control clock circuit able to support operations from 75 to 19,200 bps asynchronous, or 250 to 500 K bps in a synchronous mode. All 16 ports can be jumpered to be paged at the same address via Ohio Scientific's memory management hardware. Each port is based on a fully programmable ACIA capable of running both in asynchronous or fully synchronous mode. The CA10-X 16 port serial interface board is available for $200 for the first two ports plus $50 additional for each extra port up to 16.■

Circle 619 on inquiry card.

Used Computer Equipment Available

Computer TEXTile announces the availability of its line of reconditioned hard copy hardware. The company features the DTC 300A terminal which uses the Diablo Hytype I daisy wheel printing mechanism and the Gen Com 3000 terminal which uses Qume Q30 series daisy wheel printing mechanism. Both of these terminals run at 30 characters per second, are ASCII encoded and come standard with an RS-232 interface. The Diablo terminal has an 82 key keyboard with 11 key numeric pad. The Qume terminal has an 86 key keyboard with a 15 key numeric pad. Both terminals have full graphics capability. The Qume terminal also features Super Plot which allows plotting at 5 to 50 times the speed obtainable with the Diablo.

Options include 45 and 55 characters per second mechanisms for some Qumes, pin feed platens and tractor form feeders. Both terminals come with a limited 30 day warranty.

The Diablo is priced at $1995 and the Qume is $2150. Computer TEXTile is located at 10960 Wilshire Blvd, Suite 1504, Los Angeles CA 90024.■

Circle 617 on inquiry card.

Alphanumeric Display and Keyboard Controller

The MTX-B1 is a general purpose programmable alphanumeric display and keyboard interface device for use with an 8 bit processor. The display portion provides all timing and refresh signals to drive up to 32 characters in standard 7, 14 or 16 segment formats. The keyboard portion provides all scanning signals, and debounces and decodes any keyboard with up to 64 keys. The single integrated circuit controller interfaces directly to the microprocessor via the data bus. Many intelligent commands for display and keyboard manipulation are incorporated.

The integrated circuit requires a single +5 V (±10%) 60 mA power supply. All display and keyboard input/output (IO) pins are TTL compatible. The MTX-B1 can be interfaced to any TTL, CMOS, or NMOS processor through an IO port or bus. The integrated circuit is available in a 40 pin dual-in-line package (DIP) for $49. For more information write Matrox Electronic Systems, POB 56, Ahuntsic Sta, Montreal Quebec H3L 3N5 CANADA.■

Circle 618 on inquiry card.

Visit our new retail location!





SAVE 10%

LM3A 3 dig 1% DC $120.60
LM3.5A 3½ dig .5% DC $142.65
LM40A 4 dig .1% DC $188.10
LM4A 4 dig .03% DC $225.00

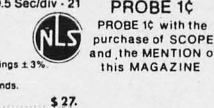
- Rechargeable batteries and charger included
- Measures DC Volts, AC Volts, Ohms and Current
- Automatic polarity, decimal and overload indication
- Measures DC Volts, AC Volts, Ohms and Current
- Automatic polarity, decimal and overload indication
- No zero adjustment and no full-scale ohms adjust
- Battery-operated — NiCad batteries; also AC line operation.
- Large LED display for easy reading without interpolation
- Size: 1.9"H x 2.7"W x 4"D
  Parts & labor guaranteed 1 year
  Tilt stand option .... $ 3.50
- Leather case $20.00



$318.
MS-15 MINISCOPE

With Rechargeable Batteries & Charger Unit

- 15 megahertz bandwidth.
- External and internal trigger.
- Time base — .1 microsec. to 0.5 Sec/div - 21 settings . ± 3%.
- Battery or line operation.
- Automatic & line sync modes.
- Power consumption < 15 watts.
- Vortical Gain — .01 to 50 V/div - 12 settings ± 3%.
- Viewing area 1.1" x 1.35".
- Case size 2.7"H x 6.4"W x 7.5"D, 3 pounds.
- Parts & Labor guaranteed 1 year
- 10 to 1, 10 meg probe .... $ 27.
- Leather carrying case .... $ 45.



PROBE 1¢
PROBE 1¢ with the purchase of SCOPE and the MENTION of this MAGAZINE

- **MS-215 Dual Trace Version of MS-15 $435.**

SALE S-100 BUS EDGE CONNECTORS SALE

S100-WWG 50/100 Cont 125 ctrs 3 LEVEL WIRE WRAP 025" sq posts on 250 spaced rows GOLD plated

1-4	5-9	10-24
$4.00	$3.75	$3.00

S100-STG 50/100 Cont. 125 ctrs. DIP SOLDER TAIL on 250 spaced rows for VECTOR and MASI motherboards GOLD plated

1-4	5-9	10-24
$3.50	$3.25	$3.00

RG81G 50/100 Cont 125 ctrs DIP SOLDER TAIL on 140 spaced rows for ALTAIR motherboards GOLD plated.

1-4	5-9	10-24
$5.00	$4.75	$4.50

R681-3 50/100 Cont 125 ctrs. PIERCED SOLDER EYELET tails. GOLD
$7.53

Other Popular Edge Connectors

R644-G 22/44 Cont. 156 ctrs. PIERCED SOLDER EYLET tails GOLD plated

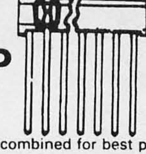
1-4	5-9	10-24
$3.00	$2.75	$2.50

R644-3 22/44 Cont. 156 ctrs. WIRE WRAP tails GOLD.

1-4	5-9	10-24
$4.00	$3.75	$3.50

CG-1 IMSAI Style Card Guides 5/$1.00
ATTN: OEM's and Dealers, many other connectors available call or quotation.

3 LEVEL GOLD WIRE WRAP SOCKETS



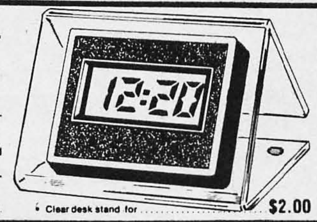
Sockets purchased in multiples of 50 per type may be combined for best price.

	1-24	25-49	50-99	100-249	250-999	1K-5K
8 pin*	.41	.38	.35	.31	.27	.23
14 pin*	.39	.38	.36	.32	.29	.27
16 pin*	.43	.42	.39	.35	.32	.30
18 pin	.63	.58	.54	.47	.42	.36
20 pin	.80	.75	.70	.63	.58	.53
22 pin*	.90	.85	.80	.70	.61	.57
24 pin	.90	.84	.78	.68	.63	.58
28 pin	1.10	1.00	.90	.84	.76	.71
40 pin	1.50	1.40	1.30	1.20	1.04	.89

All sockets are GOLD 3 level closed entry *End and side stacable. 2 level, Solder Tail, Low Profile, Tin Sockets and Dip Plugs available. CALL FOR QUOTATION

LIQUID CRYSTAL DIGITAL CLOCK-CALENDAR

- For Auto, Home, Office
- Small in size (2x2½x½)
- Push button for seconds release for date.
- Clocks mount anywhere with either 3M double-sided tape or VELCRO, included.
- 2 MODELS AVAILABLE:
  LCD-101, portable model runs on self-contained batteries for better than a year.
  LCD-102, runs on 12 Volt system and is back-lighted.
- LCD-101 or LCD-102 your choice.................. **$34.95** ea

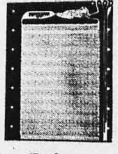


- Clear desk stand for **$2.00**

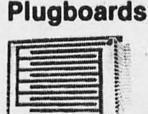
- Kit includes 12 tantalum capacitors for +5, +12, —12 buses and insulated mounting spacers.
- Wiring side shown. Component side bare epoxy glass with white markings for component locations.
- G10 epoxy glass board with 2 ounce copper, solder plated and 038 diameter holes for leads.
- Solder mask with solder windows on etched circuits to avoid accidental short circuits.
- Mounts 11 receptacles with 100 contacts (2 rows) on 125 centers with 250 row spacing, Vector part number R681-2, or mounts 10 receptacles plus interconnections to smaller mother board for expansion.
- Includes etched circuits and instructions for option of active, pull-up, or floating terminations.
- Large buses: +5V and GND (10 AMPS), ± 12V or 16V (7 AMPS). Current ratings are per MIL-STD-275 with 10°C rise.
- Fits in Vector-pak enclosures.
- Fits in IMSAI 8080 microcomputer as expander board.

8803 MOTHER BOARD FOR S100 BUS MICRO-COMPUTERS

Vector



Price: $29.50



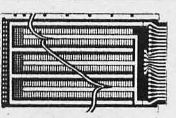
Vector **Plugboards**

8800V
Universal Microcomputer/processor plugboard, use with S-100 bus. Complete with heat sink & hardware 5.3" x 10" x 1/16"

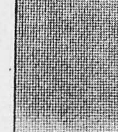
1-4	5-9	10-24
$19.95	$17.95	$15.96

8801-1
Same as 8800V except plain; less power buses & heat sink.

1-4	5-9	10-24
$14.95	$13.46	$11.96

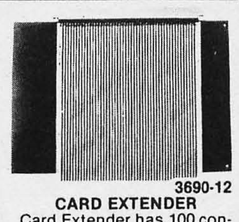


3682 9.6" x 4.5"
$10.97
3682-2 6.5" x 4.5"
$9.81
Hi-Density Dual-In-Line Plugboard for Wire Wrap with Power & Grd. Bus Epoxy Glass 1/16" 44 pin con. spaced .156

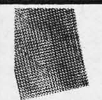


3677 9.6" x 4.5"
$10.90
3677-2 6.5" x 4.5"
$9.74
Gen. Purpose D.I.P. Boards with Bus Pattern for Solder or Wire Wrap. Epoxy Glass 1/16" 44 pin con. spaced .156

3662 6.5" x 4.5"
$7.65
3662-2 9.6" x 4.5"
$11.45
P pattern plugboards for IC's Epoxy Glass 1/16" 44 pin con. spaced .156



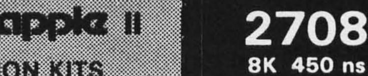
3690-12
CARD EXTENDER
Card Extender has 100 contacts 50 per side on .125 centers-Attached connector-is compatible with S-100 Bus Systems. **$25.83**
3690 6.5" 22/44 pin .156 ctrs. Extenders $13.17



1/16 *Vector* BOARD
.042 dia holes on 0.1 spacing for IC's

		PRICE	
Phenolic			
PART NO.	SIZE	1-9	10-19
64P44XXXP	4.5x6.5"	$1.56	$1.40
169P44XXXP	4.5x17"	$3.69	$3.32
Epoxy Glass			
64P44	4.5x6.5"	$1.79	$1.61
84P44	4.5x8.5"	$2.21	$1.99
169P44	4.5x17"	$4.52	$4.07
169P84	8.5x17"	$8.03	$7.23

TRS-80



MEMORY EXPANSION KITS
4116's RAMS
(16Kx1 300ns)
8 for $79.95
add $5.00 for new Programming. Jumpers for TRS-80 may also be used for S.D. EXPANDORAM

2708
8K 450 ns
EPROM
FACTORY PRIME
$ 9.00 EA.
25 + *Call For Price*



14 & 16 PIN GOLD 3 LEVEL WIRE WRAP SOCKETS
14 - G3 100 for **$30.00**
16-G3 100 for **$30.00**
50 of each for **$32.00**
Sockets are End & Side stackable, closed entry

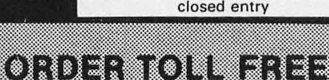




LEDU

IM-10A List $85.00
SPECIAL
$54.95 with tube

Perfectly balanced fluorescent lighting with precision magnifier lens. Tough thermoplastic shade. Easy lens removal. New wire clip design permits easy installation and removal of fluorescent tube. Comes with plastic shield to protect tube from soiling and damage.
Colors: Gray, Black, and Chocolate Brown. Comes with one 22 watt T-9 Circline fluorescent tube. 3 diopter lens.



ORDER TOLL FREE
1-800-423-5633
except CA., AK., HI., Call
(213) 894-8171



Price Breakthrough!
$16.95 or 3/$45.00



MA1003 CAR CLOCK
Bright Green Fluorescent Display Crystal Time Base Assembled, just add switches and 12 VDC.

SPECIAL
14CS2 100 for $14.00
16CS2 100 for $16.00
14 pin CS2 10 for $2.00
16 pin CS2 8 for $2.00
These low cost DIP sockets will accept both standard width plugs and chips.
For use with chips, the sockets offer a low profile height of only .125" above the board. These sockets are end stackable.



WRAP POST
for .042 dia. holes (all boards on this page)
T44/C pkg. 100 .. **$ 2.34**
T44/M pkg. 1000 ..... **$14.35**
A-13 hand installing tool .... **$ 2.94**



PRIORITY ONE ELECTRONICS ©
16723B Roscoe Blvd. Sepulveda, CA 91343

VISA — master charge

Terms: VISA, MC, BAC, check, Money Order, C.O.D., U.S. Funds Only. CA residents add 6% sales tax. Minimum order $10.00. Orders less than $75.00 include 10% shipping and handling; excess refunded. Just in case . . . please include your phone no. **Good thru FEB. '79**

Send for our latest brochure. **phone orders welcome (213) 894-8171** OEM and Institutional inquiries invited.

24 PIN DIP PLUGS WITH COVERS



3 / $1.00
40 / $10.00

Visit our new retail location! Visit our new retail location!
Visit our new retail location! Visit our new retail location!
Visit our new retail location!

What's New?

New Kit Makes a "Flippy Disk" Out of a Floppy Disk

Called the Flippy Disk Kit, this kit contains all the necessary tools to locate and accurately punch extra holes in a 5½ inch disk so that the spare side of the disk can be used. Instructions explain the function of each hole and opening in the jacket and the method of marking and punching the holes and testing the newly available side. The kit is designed to be used with North Star, Horizon, PolyMorphic, Vector Graphic, Vista or any other 5¼ inch hard sectored disk drive. The kit contains instructions, double sided "flippy plate," a pencil for making highly visible marks on the black diskette jacket, and a ground and polished hand punch for making the holes. The kit is priced at $9.95 plus $1 for shipping and is available from Square 1, 614 18th Av, Menlo Park CA 94025. ■

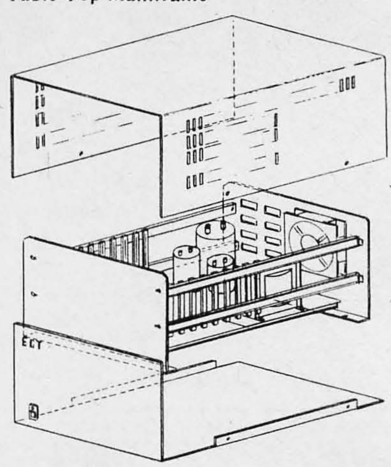
Circle 525 on inquiry card.

Data Base Management System for Small Business Applications

Ohio Scientific has introduced a programming breakthrough for small business microcomputer applications. This programming innovation is called OS-DMS, Ohio Scientific's data base management system. This system brings the use of microcomputers down to the level of nonprogrammers. It allows untrained computer users to store and recall information from any one of Ohio Scientific's full line of floppy and hard disk microcomputer systems. Each program in the OS-DMS library is aimed at a specific small business application, such as real estate, automotive sales, mailing lists, inventory, accounts receivable and accounts payable, inventory and invoice, ledger, personnel files, retail sales, medical files, etc. Because the microcomputer uses terms familiar to each specific application, the user doesn't have to learn programming languages. OS-DMS data base management system allows knowledgeable computer operators to fully utilize the computer's resources and to perform tasks routinely which usually require programming on nondata base systems. Each program is priced at $300. For further information, write to Ohio Scientific Inc, 1333 S Chillicothe Rd, Aurora OH 44202. ■

Circle 526 on inquiry card.

Table Top Mainframe



The TT-10 Table Top Mainframe consists of an industrial quality card cage; the MB-10, an S-100 bus mother board with bus termination and ground plane to reduce noise; a full set of ten connectors and guides; a power supply which mounts inside the card cage and furnishes 15 A at 8 V, 1.5 A at +16 V and 1.5 A at −16 V; a clear satin finished front and bottom plate with a reset switch and power indicator light emitting diode; and a whisper fan and cover.

The TT-10-K kit is $325, the TT-10-A assembled is $395. Contact Electronic Control Technology, 763 Ramsey Av, Hillside NJ 07205. ■

Circle 527 on inquiry card.

Circle 115 on inquiry card.

California Digital

Post Office Box 3097 B • Torrance, California 90503



Hazeltine 1400

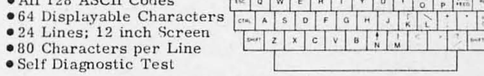
cost effective

CRT TERMINAL

$735 plus shipping

The Hazeltine 1400 Vidio Display Terminal is designed to optimize interactive real-time operations. The interface is capable of either local or remote connection through an EIA RS232-C interface at baud rates that are switch selectable up to 9600 baud.

- All 128 ASCII Codes
- 64 Displayable Characters
- 24 Lines; 12 inch Screen
- 80 Characters per Line
- Self Diagnostic Test



Immediate Delivery

S-100 Mother Board

Quiet Buss

$29.95

8803-18

18 slot

IMSAI

The Quiet Buss from California Industrial is quality engineered. No short cuts have been taken to produce this mother board. Active termination circuitry prevents noise and crosstalk. Manufactured from extra heavy FR-4 epoxy glass.

TELETYPE® MODEL 43



Even if we have to give them away, we're going to ship more 43's in 1979 than the aggregate of all our competitors.

Model 43AAA (TTL)

EACH	3	10	25
$925.	875.	850.	825.

plus shipping

RS-232 Interface "K" Add $75.00

Shugart Associates

SA800-R Floppy Disk Drive

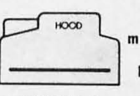
The most cost effective way to store data processing information, when random recall is a prime factor. The SA800 is fully compatible with the IBM 3740 format. Write protect circuitry, low maintenance & Shugart quality.

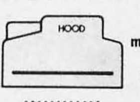
$449.50

$29.95 BOX of 10 DISKETTES

Verbatim. APPLE/TRS-80 Mini - Soft sector

CONNECTORS





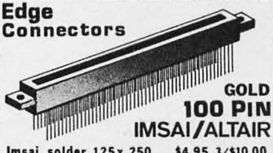
your choice

DB25P male plug & hood or DB25S female

$3.95

Qty.	fe.	male	hd.
10	3.45	2.45	1.15
25	3.15	2.25	1.05
100	2.85	1.90	.95
500	2.25	1.60	.85
1K	1.97	1.37	.73

Edge Connectors



GOLD 100 PIN IMSAI/ALTAIR

Imsai solder .125x.250	$4.95	3/$10.00
Imsai w/w .125centers	$4.95	3/$13.00
Altair soldertail .140 row	$5.95	3/$15.00
SPECIALS		
22/44 Kim eyelet .156"	$1.95	3/$5.00
25/50 solder tab .156"	$1.09	3/$2.00
36/72 wide post w/w .156	$1.95	3/$5.00

SPERRY UNIVAC KEYBOARD

$24.88

The famous Sperry Univac 1710 Hollerith keyboard assembly is now available from California Industrial for only $24.88. The ideal computer input device for accountants and mathematicians. The numeric keys are placed on the lower three rows to resemble a ten key adding machine. This format allows one handed numeric data entry.
Original cost was $385. Used but guaranteed in excellent condition. Complete with documentation.

FROM ATARI COLOR TELEVISION R.F. MODULATOR

$13.95



Assembled

The Atari R.F. Modulator allows computer data to be displayed directly upon your existing television system. This unit converts the signal from the Apple II and other video sources into television frequencies. Operates from single 5 volt supply. Complete with metal case, mating R.F. connector and 15 feet of coax cable. Schematics and instructions included.

SPECIAL APPLE II 16K MEMORY

COLOR • GRAPHICS • SOUND

$1024 PLUS SHIPPING

Mfg. Sug. Retail.... $1195

Scotch® BRAND

$4.98 10 for $45.

DISKETTES

8 inch Soft (IBM)
8 inch 32 sector
Mini Soft sec.
Mini 10 sector
Mini 16 sector

Certified Digital CASSETTES

Won't drop a BIT!

$5.50



CALIFORNIA INDUSTRIAL is an Authorized Dealer of Scotch Brand Data Products

APPLE RS-232

Serial Interface **$59.95**

Interfaces Apple II to Teletype or other serial printer.

ELECTRONIC SYSTEMS

dataphone® model 2400 $249.

Factory new Bell System modems. Some units show cosmetic damage. Documentation not included. Other models available call for price and quantities.

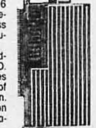
Extender Board

Mullen $34.95

Facilitates design and troubleshooting of all S-100 microsystems. Includes logic probe along with high-low and pulse LED display.
Also available, the Mullin CB-1 controller board $88.

S-100 PROTOTYPE BOARD

$19.98



GP100-Maximum design versatility along with standard address decoding and buffering for S100 systems. Room for 32 uncommitted 16 pin IC's, 5 bus buffer & decoding chips, 1 DIP address select switch, a 5 volt regulator and more.
WW100-Wire wrap breadboard, similar to the GP100. Allows wire wrap of all sizes of sockets in any sizes of sockets in any combination. An extra regulator position for multiple voltage applications.

MEMORY

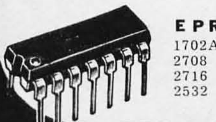
DYNAMIC	1-7	8-32	32+
4115 8Kx1	11.95	11.50	*
4116 16Kx1 (Apple II & TRS80)	13.95	13.00	12.25
4164 64Kx1	*	*	*

* As you may be aware, publishers require advertisers to submit their ad copy 60 to 90 days prior to "press" date. That much lead time in a volatile market place, such as memory circuits, makes it extremely difficult to project future cost and availability.
To obtain the best pricing on memory we have made volume commitments to our suppliers, which in turn affords us the opportunity to sell these circuits at the most competitive prices. Please contact us if you if you have a demand for volume state of the art memory products.

STATIC	1-31	32-99	100-5C	-999	1K+
21L02 450nS.	1.49	1.19	1.05	.95	.89
21L02 250nS.	1.69	1.49	1.45	*	*
2114 1Kx4 450	6.95	6.50	6.25	6.00	5.75
2114 1Kx4 300	8.95	8.50	8.00	*	*
4044 4Kx1 450	8.95	8.50	8.00	*	*
4044 4Kx1 250	9.95	9.50	9.00	*	*
4045 1Kx4 450	8.95	8.50	8.00	*	*
4045 1Kx4 250	9.95	9.50	9.00	*	*
5257 low pow.	7.95	7.50	7.05	6.75	6.45

SPECIAL CIRCUITS

Z80A 4 MHz.	24.95	AY5-1013A UART	4.95
8080A CPU	9.95	Floppy Disc Controllers	
8085	22.50	WD 1771 single D.	39.95
8086 Intel 16 bits	*	WD 1781 Double D	65.00
TMS 9900 16 bits	49.95	WD 1791 D/D 3740	*



EPROMS		1-15	16-63	64+
1702A	2K	4.95	4.50	4.00
2708	8K	9.95	9.50	9.00
2716	16K	19.95	*	*
2532	32K	*	*	*

CLARE-PENDAR General Instrument Corp. KEYBOARD ASCII ENCODED



This is a one time purchase of NEW Surplus keyboards, recently aquired from the Telecommunications Division of the Singer Corporation.
The keyboard features 128 ASCII characters in a 63 key format, MOS encoder circuitry "N" key rollover, lighted shift lock, control, escape and repete functions.
Sloped pannel and positive feel switches, makes this professional quality keyboard an excellent buy at only $64.95.
Limited Quantities.

63 KEY **$64.95**

Thumbwheel switch



Ten position BCD

$1.39 ea.

10	50
$1.19	.89

MINIATURE SWITCHES

your choice

	10	50	100	1k
$.98	$.88	.81	.73	.66

SPDT Miniature Toggles

7101 C&K	ON-NONE-ON
7107 jbt	ON-OFF(mnt.ON)
7108 CK	ON-(moment. ON)
Rocker JBT	DPDT
Rotary	3P-4-Pos.
Rotary	3P-6-Pos.
Push B (N.O.)	$.39ea. 4/$1

CAPACITORS

ELECTROLYTICS

	ea.	10	50
80,000/10v.	3.95	3.49	2.95
4500/50v.	$1.49	1.35	1.19
1000/15v axial	$.55	49	.45
.1 disc	$.12	.09	.07
.01 disc	.06	.05	.04



DIP Switch

$1.49 ea.

10	25	100	1K
$1.29	1.15	.97	.83

specify 7 or 8 pos.

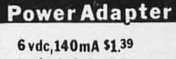
SPST

50 Conductor RIBBON WIRE TWISTED PAIR

$.69 Ft.

Power Adapter

6 vdc, 140mA	$1.39
7 vdc, 1.4 A.	5.50
9 vdc 200mA.	1.19
10 vAc 300mA.	1.95



SPECIAL SPECIAL SPECIAL

Only 10,000 Available

NE555H

Leads fan out to fit Mini-DIP socket.

3 for $.98

25 for $7.00 • 100 $19.50

Transistors

	ea.	10	50	100
2N2222A	.20	.18	.16	.15
2N3055	.69	.65	.59	.55
MJ3055	.79	.75	.69	.65
2N3772	1.59	1.49	1.39	1.29
2N3904	.15	.11	.09	.07
2N3906	.15	.11	.09	.07

Diodes

		10	25	100
1N4002	100v.	.08	06	.05
1N4005	600v.	.10	.08	.07
1N4148	signal	.07	.05	.04

jumbo red **LED's**

ea.	10	25	100
$.15	.13	.11	.09



$.98

Output: 12v.ct. 175mA. TRANSFORMER

TRIMMER POTENTIOMETERS

2K 5K 10K 50K

5 for $.98

20	50	100
16¢	14¢	12¢

9 foot $1.49



Heavy duty grounded power cord and mating chassis connectors.

Page Wire Wrap Kits precut & stripped

KIT No.1 $6.95 900 Assorted Lengths

KIT No.2 $19.95 2800 Assorted Lengths 250' Bulk Wire

DISCOUNT Wire Wrap Center

IC SOCKETS

pin	wire wrap ea.	25	50	low profile ea.	25	50
8				17¢	16	15
14	37¢	36	35	18	17	16
16	38	37	36	19	18	17
24	99	93	85	36	35	34
40	169	155	139	63	60	58

50ft. $.98 KYNAR WIRE WRAP

500	1,000	11,000
$9.	$15.	$105.

Battery wire wrapping tool $34.95

$29.95 BW630

OK HOBBY WRAP-30 wire wrap & strip tool $5.45

(213) 679-9001



master charge

All merchandise sold by California Industrial is premium grade.
Orders are shipped the same day received.
PLEASE INCLUDE $1.00 SHIPPING ON ORDERS UNDER $15.00
California residents add 6% sales tax • Money back guarantee.
Sorry, no COD's • Foreign orders add 10%
Credit cards accepted, $20 minimum

Circle 39 on inquiry card.

Circle 230 on inquiry card.



FORMERLY CYBERCOM/SOLID STATE MUSIC.

CB-1 8080 Processor Board. 2K of PROM 256 BYTE RAM power on/rest Vector Jump Parallel port with status. Kit ....$125.95 PCBD $28.95

MB-6A Basic 8KX8 ram uses 2102 type rams, S-100 buss. Kit 450 NSEC ......$123.95. PCBD......$24.95

MB-7 16KX8, Static RAM uses μP410 Protection, fully buffered KIT................$299.95

MB-8A 2708 EROM Board, S-100, 8K8X or 16KX8 kit without PROMS $75.00 .... PCBD $28.95

MB-9 4KX8 RAM/PROM Board uses 2112 RAMS or 82S129 PROM kit without RAMS or PROMS $72.00

IO-2 S-100 8 bit parallel /IO port, ⅔ of boards is for kludging. Kit ....$46.00 PCBD......... $26.95

IO-4 Two serial I/O ports with full handshaking 20/60 ma current loop: Two parallel I/O ports. Kit ....$130.00 PCBD $26.95

VB-1B 64 x 16 video board, upper lower case Greek, composite and parallel video with software, S-100. Kit................$125.00 PCBD.............. $26.95

Altair Compatible Mother Board, 11 x 11½ x ⅛". Board only ......$39.95. With 15 connectors......$94.95

Extended Board full size. Board only .... $ 9.49
With connector $13.45

SP-1 Synthesizer Board S-100
PCBD.............. $42.95 KIT................$135.95

		PRIME SUPPORT	
82S23	$1.50	8080A	$ 9.95
82S123	1.50	8212	3.25
82S126	1.95	8214	6.50
82S129	1.95	8224	3.49
82S130	3.95	L2114 (450 NSEC)	7.25
82S131	3.95	L2114 (250 NSEC)	7.99
MMI6330	1.50	2102AL-2	1.70
IM5600	1.50	21L02 (450)	1.25
IM5603	1.95	2708Q (National)..	8.95
IM5604	3.95	4116 (Apple Ram)	14.99
IM5610	1.50		
IM5623	1.95		
IM5624	3.95		

WMC inc. **WAMECO INC.**

FDC-1 FLOPPY CONTROLLER BOARD will drive shugart, pertek, remic 5" & 8" drives up to 8 drives, on board PROM with power boot up, will operate with CPM (not included). DEC. & JAN. ONLY SPECIAL PCBD $23.95

FPB-1 Front Panel. (Finnally) IMSAI size hex displays. Byte or instruction single step. PCBD special, DEC. & JAN. $39.95

MEM-1 8KX8 fully buffered, S-100, uses 2102 type rams PCBD $24.95

QM-12 MOTHER BOARD, 13 slot, terminated, S-100 board only $34.95

CPU-1 8080A Processor board S-100 with 8 level vector interrupt PCBD $25.95

RTC-1 Realtime clock board. Two independent interrupts. Software programmable. PCBD $25.95

EPM-1 1702A 4K Eprom card PCBD $25.95

EPM-2 2708/2716 16K/32K
EPROM CARD PCBD $24.95

QM-9 MOTHER BOARD. Short Version of QM-12. 9 Slots PCBD $30.95

MEM-2 16K x 8 Fully Buffered
2114 Board PCBD $25.95

16K RAM BOARD by HWE fully buffered, bank select standard to IEE buss gold fingers, solder mask, plated thru holes, silk screened ....PCBD $25.95

KLUDGE BOARD by HWE for S-100 glass epoxy over 2600 plated through holes, 4 regulators with CAPS all S-100 functions labeled, gold fingers. PCBD $29.95

MIKOS

419 Portofino Drive
San Carlos, California 94070
Please send for IC, Xistor and Computer parts list.

1ST OF THE YEAR SALE ON PREPAID ORDERS

(charge cards not included on this offer)

8K x 8 RAM. Fully buffered, assembled with sockets, tested or burned in. Part may be house numbered. 450 NSEC. Limited quantity. $99.99
4K x 8 EPROM. Fully buffered with Intel 1702A. Kit $74.86

MIKOS PARTS ASSORTMENT WITH WAMECO AND CYBERCOM PCBDS

MEM-2 with MIKOS #7 16K ram with L2114 450 NSEC $249.95
MEM-2 with MIKOS #13 16K ram with L2114 250 NSEC $269.95
MEM-1 with MIKOS #1 450 NSEC 8K RAM $123.95
CPU-1 with MIKOS #2 8080A CPU 89.95
MEM-1 with MIKOS #3 250 NSEC 8K RAM 144.95
QM-12 with MIKOS #4 13 slot mother board 89.95
RTC-1 with MIKOS #5 real time clock 60.95
VB-1B with MIKOS #6 video board less molex connectors 99.95
EMP-1 with MIKOS #10 4K 1702 less EPROMS 49.95
EPM-2 with MIKOS #11 16-32K EPROMS less EPROMS 49.95
QM-9 with MIKOS #12 9 slot mother board 67.95

MIKOS PARTS ASSORTMENTS ARE ALL FACTORY PRIME PARTS. KITS INCLUDE ALL PARTS LISTED AS REQUIRED FOR THE COMPLETE KIT LESS PARTS LISTED. ALL SOCKETS INCLUDED.

VISA or MASTERCHARGE. Send account number, interbank number, expiration date and sign your order. Approx. postage will be added. Check or money order will be sent post paid in U.S. If you are not a regular customer, please use charge, cashier's check or postal money order. Otherwise there will be a two-week delay for checks to clear. Calif. residents add 6% tax. Money back 30 day guarantee. We cannot accept returned IC's that have been soldered to. Prices subject to change without notice. **$10 minimum order. $1.50 service charge on orders less than $10.00.**

ATTENTION TRS-80 & APPLE USERS

A PRINTER FOR YOUR COMPUTER



$995.00
MODEL 3S-80 for TRS-80
Ready to plug into your expansion interface.

MODEL 3S-PP
for computers with 8 bit serial port.

MODEL 3S-SS
for computers with RS-232 port.

$1095.00 for
MODEL 3S-AA
Includes RS-232 card for AppleII
Specify model number on order.

- Ready to plug into your computer
- Very high quality print
- Completely refurbished IBM 731 I/O Selectric terminal in a new table
- Upper & lower case removable type ball
- Special I/O interface
- Heavy duty re-mfg. IBM power supply

TERMS: VISA, MASTERCHARGE, Cashier Check or Money Order. C.O.D. with 10% down. Shipping Via Air or Truck collect.

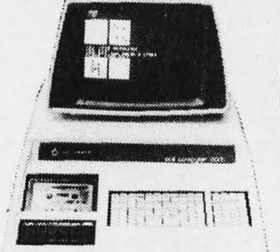
3 S SALES
P.O. BOX 45944
TULSA, OK 74145
918/622-1058

Circle 379 on inquiry card.

BUY FROM THE BIGGEST

NCE/CompuMart

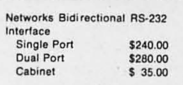
1250 North Main Street, Department **BY19**
P.O. Box 8610 Ann Arbor, Michigan 48107



PET

$795

THE PET CONNECTION



Networks Bidirectional RS-232 Interface	
Single Port	$240.00
Dual Port	$280.00
Cabinet	$ 35.00
Networks Modem for PET	
Attaches to DAA	$320.00
Cabinet	$ 35.00
PET with 8K user memory	$795.00
Second Cassette drive	$99.95
RS-232 Printer Adapter — assembled board	$98.50
With power supply, case & connector	$169.00
Beeper — PET controlled 'beep'	$24.95
Video Buffer — add a conventional monitor	$29.95
EXPANDAPET - Economical 16K memory expansion for the PET (includes 1 daughter board)	$435.00
without daughter board	$395.00

Hams! Add a PET to your rig.

Computerize your station with Microtronics' M-65 Ham Interface and included software. You simply plug the unit into your PET and you're ready to display, transmit and receive either CW or RTTY at rates up to 100 WPM.

ASSEMBLED KITS

$99.95

SECOND CASSETTE FOR PET-$99.95

This peripheral from Commodore gives you the ability to use one tape deck for input and one for output, copy files, etc.

SELECTRIC-BASED TERMINAL FOR PET

$895

We have interfaced our TC-71 to the PET. Use with the IEEE-488 bus as a printer or with the VIA User Port for bidirectional I/O. Ideal for Word Processing applications.

Sale price if bought with PET: $795

LOW-COST PRINTER FOR PET

This is a completely refurbished Teletype KSR-33 terminal ready to attach to your PET's interface to use as a printer.

INTRODUCTORY PRICE $395

READER/PUNCH

8 LEVEL PAPER TAPE/READER PUNCH

USED (90 day warranty) $175.00
Documentation Package $15.00

NEW ASCII INTERFACES for Reader/Punch

30 Day delivery on Interfaces

Boards & Cables	
Parallel	$220
RS232 Serial	$350
Board, Cables, Power Supply	
Parallel	$490
Serial	$615
Board, Cables, Power Supply, Enclosure	
Parallel	$575
Serial	$780

EXIDY SORCERER $895

- Z-80 Processor
- Full-size ASCII keyboard
- User definable characters
- Up to 32K on-board RAM
- Interchangeable ROM PACs
- 30 x 64 display
- Resident 4K Monitor ROM
- Dual Cassette I/O
- Serial and Parallel I/O
- Complete Operators Manual
- Only $895

Top value in our catalog, simply add a monitor and tape decks to complete the system. Can be used as a terminal. Later ROM PACs will offer Z-80 Assembly Language, APL, PILOT, Word Processing and a DOS PAC for disk-based FORTRAN and COBOL. What a system for $895

Sorcerer 8K	$895
Sorcerer 16K	$1150
Sorcerer 32K	$1395
16K Expansion	$299

Single board
VIDEO TERMINAL by XITEX

SALE!

16 Ln x 64 chr. display. 128 printable characters, software UART (ASCII or Baudot). 3 serial interfaces & cassette I/O, true composite video, TTL compatible, easy power connection.

Assembled & tested **NOW $187**



Hazeltine 1400

IN STOCK NOW

Only $749.00

Now a 5th Generation in Terminals:

Introductory Sale Price — Immediate Delivery

- 24 x 80 display
- RS-232C interface
- 110 to 9600 Baud
- 12" CRT Monitor
- Remote screen control
- Uses 8048 microprocessor
- State-of-the-art design

A full capability terminal completely assembled with a 90 day warranty for less than you'd pay for a kit.

KIM-1

The original fully expandable single board computer

Assembled **$179**

with power supply **$209**

EXPAND YOUR KIM SYSTEM:

KIMSI
KIM to S-100 Board

Assembled	$169
Connector set	$15
KIM4 Expansion board	$119
MICRO TECH. POWER SUPPLY Assembled	$34.95

KIM ENCLOSURE

Give your KIM a professional appearance. No alterations required, sturdy Kydex 100. Black **$23.50**

WE WANT TO BUY USED MICROCOMPUTERS

Any Quantity

Urgently needed:

- Peripherals
- Manufacturers surplus
- Closed stores stock

CASH AVAILABLE

Call Today: (313) 994-3200



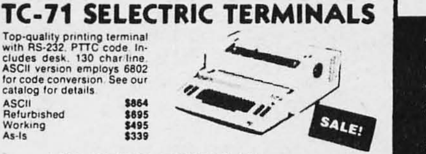
Compucolor II

NEW!

Model 3 8K RAM $1495
Model 4 16K RAM $1795
Model 5 32K RAM $2395

- Beautiful 128 x 128 color graphics
- 4K Graphics RAM
- 16K ROM with Disk BASIC
- Built-in 51K mini floppy
- RS-232 interface
- Full size keyboard
- Preprogrammed diskettes avail.

TC-71 SELECTRIC TERMINALS



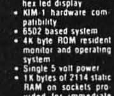


Top-quality printing terminal with RS-232, PTTC code. Includes desk, 130 char/line. ASCII version employs 6802 for code conversion. See our catalog for details.

ASCII	$864
Refurbished	$695
Working	$495
As-Is	$339

SALE!

SALE **ASCII INTERFACE FOR TC-71**
30 day delivery Normally $199 NOW **$169**
Acoustic Coupler for use with ASCII TC-71 **$159**





SYM-1 From Synertek

- 28 Double-function keypad including up to 24 "Special" functions
- Easy-to-view 6-digit hex led display
- KIM-1 hardware compatibility
- 6502 based system
- 4K byte ROM resident monitor and operating system
- Single 5 volt power
- 1K bytes of 2114 static RAM on sockets provided for immediate expansion to 4K bytes
- Enhanced software
- Standard interfaces include:
  - Audio Cassette Recorder Interface with Remote Control
  - Full Duplex 20mA Teletype Interface
  - System Expansion Bus Interface
  - TV Controller Board Interface
  - CRT Compatible Interface
- 50 I/O Lines in the basic system

$269.00

SALE ~~$225~~

SANYO MONITOR

VM4209 9" quality monitor **$149**

New Acoustic Coupler

$159

NCE/Compumart Inc. breaks the price barrier with a new 300 baud RS-232C acoustic coupler. For only $159 you get an assembled, tested stand alone unit which accepts the standard telephone handset for full or half duplex operation. Complete with power supply and carrier detect light, the unit is housed in an attractive case for desk top use.

TI PROGRAMMER

SALE!

Hexadecimal, octal, decimal calculator/converter.

Reg. $59.95 NOW **$49.95**

A Home Computer & A Video Game in a Single Unit!

INTERACT Model One



- Color graphics on your TV
- Music and sound synthesizer
- Built-in cassette tape drive
- Large 53-key keyboard
- Two joysticks included
- 14 Free cassettes worth $294
- Edu-BASIC cassette included
- 8080 Microprocessor
- 8K RAM, 2K ROM

IN STOCK NOW! **$499**

If you can't decide between a sophisticated video game and a home computer then the Interact Model One is for you. Hook it to your color TV and you've got both! You can play one of the games which are included free or you can load Edu-BASIC and write your own games. You can save programs or data using the built-in cassette tape drive. Best of all you get over $294 worth of free cassettes including:

Edu-BASIC — Write programs
Concentration — The memory game
Dogfight — Aerial battles
Add-Em Up — A math game
Biorhythm — Plot your own
Blackjack — Wager video bucks
Compute-A-Color — Easy graphics
Hangman — The word game
Knockdown — A dice game
Maze — Computer maze
Regatta — Race your ship
Showdown — Gunfight
Trailblazer — Cut off your opponent
Blank — Save programs here

Rockwell
AIM65

- Fully assembled, tested, and warranteed
- Addressing to 65K bytes (1K RAM on board)
- Includes onboard
  - Powerful 4K ROM Debug/Monitor
  - Sockets for additional 4K RAM 16K ROM
  - 20 Col. ASCII Thermal printer (90 lines/min.)
  - Full size (54 key) Alphanumeric keyboard
  - 20 char ASCII Alphanumeric display
  - Application Conn. for TTY, 2 cassettes, 2 8 big I/O ports
  - Expansion Conn. (try our KIMSI S-100 board)
  - Both connectors totally KIM-1 compatible
- Easy Power connections

$375

Options — 4K ROM Assembler/Text Editor
8K Basic ROM

10 DAY RETURN PRIVILEGE

- 90 day limited warranty
- Newest, most exciting products in personal computing
- Outstanding values
- All items fully assembled and tested unless otherwise stated.
- Send for our catalog for more detailed product information.

1-800 521 1534



Just say charge it! Which card?



• Michigan residents add 4% sales tax. • Foreign orders (except Canada) add 10% additional handling (U.S. currency only). • P.O.'s accepted from D&B rated companies. • Toll free phones open from 9:00 a.m.-7:00 p.m. EST Monday - Friday, 10:00 a.m.-5 p.m. Saturdays. • Open accounts invited — call for credit application. • Most items in stock for immediate shipment — call for delivery quotation. • Sorry — NO C.O.D.'s. • All prices subject to change without notice. • Write for our FREE catalog. • In the Ann Arbor area? Retail store open 11:00 a.m.-7:00 p.m. Tuesday-Friday, 10:00 a.m.-5:00 p.m. Saturdays.

Circle 283 on inquiry card.

Computer System from Dynabyte Offers New Disk Drive Controller Technology



The DB8/2 computer system from Dynabyte Inc, 1005 Elwell Ct, Palo Alto CA 94303, offers up to 1.2 M bytes of mass storage on two 5 inch drives. It uses 77 track Micropolis disk drives and with Dynabyte's controller offers double or quad density in single or double sided configurations. To implement the drives, a dual density floppy disk controller was developed. It is capable of handling a variety of 5 inch and 8 inch drives in dual density on either one or two sides. To permit expansion of the system as the need increases, the controller is capable of handling up to 16 drives. The dynamic data compensation yields a double density error rate comparable to single density rates.

The DB8/2 includes a 4 MHz Z-80 processor module that also contains two RS232 serial input and output (IO) ports, one parallel IO port, an erasable read only memory programmer, two TMS2716 sockets, vectored interrupts and a real time clock. The unit has 32 K bytes of programmable memory and the disk controller in a 12 slot backplane fully populated with military specification connectors. It uses a regulated power supply designed to comply with UL approved standards. Initial language and software packages from Dynabyte include BASIC, FORTRAN, COBOL, word processing, general ledger and accounts receivable.■

Circle 590 on inquiry card.

Dual Floppy, Z-80 Based Microcomputer from Vector Graphic



This new microcomputer, featuring two Micropolis quad density floppy disks and a Z-80 processor, has been introduced by Vector Graphic Inc, 790 Hampshire Rd, Westlake Village CA 91361. The Vector MZ features include the Z-80 processor with 158 BASIC machine language instructions and a minimum instruction cycle of 2 μs; an 18 slot motherboard which provides flexibility and expansion capabilities for up to 64 K byte of directly accessible memory using a parallel 8 bit word and 16 bit address; and up to 256 separate input and output devices can be addressed.

The disk drives are mounted directly to the front panel of the Vector MZ, providing instant access and ease of operation. Storage capacity for each of these 5¼ inch floppy disk drives is 315 K bytes formatted.

The Vector MZ is fully compatible with all existing Vector input and output (IO) and memory boards. Completely assembled and tested, the basic system includes: 18 slot motherboard, Z-80 processor with 4 MHz clock, two quad density Micropolis floppy disk drives, disk controller board, bit streamer IO board with one serial and two parallel ports, 32 K byte of static programmable memory, 12 K byte combination read only and programmable memory board with extended monitor, and Vector BASIC. The Vector MZ is priced at $3750.■

Circle 591 on inquiry card.

NMS Extends Its Pascal Machine Capabilities

A new peripherals option card for the NMS 8085A processor based microcomputer system is now available from Northwest Microcomputer Systems, 121 E Eleventh St, Eugene OR 97401. This new card, called the NMS 85/EX, gives users flexibility by allowing a number of options.

An arithmetic processing unit (APU) provides 32 bit fixed and floating point arithmetic, and floating point trigonometric operations, and gives the user the ability to do a 32 bit floating point divide in a minimum of 50 ms. It contains floating-to-fixed and fixed-to-floating conversions. All accesses to this unit are through conventional programmed input and output ports. The 3 MHz version is priced at $422 and the 2 MHz version is $272.

An interrupt controller featuring the Intel 8259 resolves priority among eight different interrupt levels according to software algorithms provided by the user. An interrupt selection area allows the user to configure a custom interrupt structure. The priority modes can be reconfigured dynamically at any time during the main program operation, allowing the user to define a complete interrupt structure based on the total system environment. The controller is priced at $90.

Interval timer capability is provided through the use of Intel's 8253 programmable interval timer. Each timer includes three 16 bit binary coded decimal or binary counters which are user programmable. The output of the first two counters can be used as interrupt request lines, to implement a real time clock. The output of the third timer may be used for the serial port, allowing software control of the desired data transmission rate. A second timer chip's three clocks may be used for extended (48 bit) timing functions through daisy chains or may be taken to the card's edge for interval timing or event counting. Each timer is available for $50.

The parallel input and output interface using the Intel 8255A programmable peripheral interface device provides 24 signal lines for transfer and control of data to or from the peripheral devices. The parallel interface is available for $90.

A serial input and output interface, using Intel's 8251 USART device, provides for complete RS-232 serial data communications, including IBM Bi-Sync, when coupled with the 8255A parallel interface option and the 8253 interval timer. The unit will operate with every current serial data transmission protocol. The transmission rate, synchronous or asynchronous mode, character length, number of stop bits and choice of parity are all program selectable. The serial interface is priced at $70 plus the parallel and timer options required.■

Circle 592 on inquiry card.

7400 TTL

		SN7470N	.29		
		SN7472N	.29		
SN7400N	.16	SN7473N	.35	SN74160N	.89
SN7401N	.18	SN7474N	.35	SN74161N	.89
SN7402N	.18	SN7475N	.49	SN74162N	1.95
SN7403N	.18	SN7476N	.35	SN74163N	.89
SN7404N	.18	SN7479N	5.00	SN74164N	.89
SN7405N	.20	SN7480N	.50	SN74165N	.89
SN7406N	.29	SN7482N	.99	SN74166N	1.25
SN7407N	.29	SN7483N	.59	SN74167N	1.95
SN7408N	.20	SN7485N	.79	SN74170N	1.59
SN7409N	.20	SN7486N	.35	SN74172N	6.00
SN7410N	.18	SN7489N	1.75	SN74173N	1.25
SN7411N	.25	SN7490N	.45	SN74174N	.89
SN7412N	.25	SN7491N	.59	SN74175N	.79
SN7413N	.40	SN7492N	.43	SN74176N	.79
SN7414N	.70	SN7493N	.43	SN74177N	.79
SN7416N	.25	SN7494N	.65	SN74179N	1.95
SN7417N	.25	SN7495N	.65	SN74180N	.79
SN7420N	.20	SN7496N	.65	SN74181N	1.95
SN7421N	.29	SN7497N	3.00	SN74182N	.79
SN7422N	.39	SN74100N	.89	SN74184N	1.95
SN7423N	.25	SN74107N	.35	SN74185N	1.95
SN7425N	.29	SN74109N	.59	SN74186N	9.95
SN7426N	.29	SN74116N	1.95	SN74188N	3.95
SN7427N	.25	SN74121N	.35	SN74190N	1.25
SN7429N	.39	SN74122N	.39	SN74191N	1.25
SN7430N	.20	SN74123N	.49	SN74192N	.79
SN7432N	.25	SN74125N	.49	SN74193N	.79
SN7437N	.25	SN74126N	.49	SN74194N	.89
SN7438N	.25	SN74132N	.75	SN74195N	.69
SN7439N	.25	SN74136N	.75	SN74196N	.89
SN7440N	.20	SN74141N	.79	SN74197N	.89
SN7441N	.89	SN74142N	2.95	SN74198N	1.49
SN7442N	.49	SN74143N	2.95	SN74199N	1.49
SN7443N	.75	SN74144N	2.95	SN74S200	4.95
SN7444N	.75	SN74145N	.79	SN74251N	1.79
SN7445N	.75	SN74147N	1.95	SN74279N	.79
SN7446N	.69	SN74148N	1.29	SN74283N	2.25
SN7447N	.59	SN74150N	.89	SN74284N	3.95
SN7448N	.79	SN74151N	.59	SN74285N	3.95
SN7450N	.20	SN74152N	.59	SN74365N	.69
SN7451N	.20	SN74153N	.59	SN74366N	.69
SN7453N	.20	SN74154N	.99	SN74367N	.69
SN7454N	.20	SN74155N	.79	SN74368N	.69
SN7459A	.25	SN74156N	.79	SN74390N	1.95
SN7460N	.20	SN74157N	.65	SN74393N	1.95

20% Discount 100 pcs combined order 25% -1000 pcs combined order

C/MOS

CD4000	.23			CD4070	.55
CD4001	.23			CD4071	.23
CD4002	.23	CD4028	.89	CD4072	.49
CD4006	1.19	CD4029	1.19	CD4076	1.39
CD4007	.25	CD4030	.49	CD4081	.23
CD4009	.49	CD4035	.99	CD4082	.23
CD4010	.49	CD4040	1.19	CD4093	.99
CD4011	.23	CD4041	1.25	CD4098	2.49
CD4012	.25	CD4042	.99	MC14409	14.95
CD4013	.39	CD4043	.89	MC14410	14.95
CD4014	1.39	CD4044	.89	MC14411	14.95
CD4015	1.19	CD4046	1.79	MC14419	4.95
CD4016	.49	CD4047	2.50	MC14433	19.95
CD4017	1.19	CD4048	1.35	MC14506	.75
CD4018	.99	CD4049	.49	MC14507	.99
CD4019	.49	CD4050	.49	MC14562	14.50
CD4020	1.19	CD4051	1.19	MC14583	3.50
CD4021	1.39	CD4053	1.19	CD4508	3.95
CD4022	1.19	CD4056	2.95	CD4510	1.39
CD4023	.23	CD4059	9.95	CD4511	1.29
CD4024	.79	CD4060	1.49	CD4515	2.95
CD4025	.23	CD4066	.79	CD4518	1.29
CD4026	2.25	CD4068	.39	CD4520	1.29
CD4027	.69	CD4069	.45	CD4566	2.25

74C00

74C00	.39			74C163	2.49
74C02	.39			74C164	2.49
74C04	.39	74C85	2.49	74C173	2.60
74C08	.49	74C90	1.95	74C192	2.49
74C10	.39	74C93	1.95	74C193	2.49
74C14	1.95	74C95	1.95	74C195	2.49
74C20	.39	74C107	1.25	74C922	5.95
74C30	.39	74C151	2.90	74C923	6.25
74C42	1.95	74C154	3.00	74C925	8.95
74C48	2.49	74C157	2.15	74C926	8.95
74C73	.89	74C160	2.49	80C95	1.50
74C74	.89	74C161	2.49	80C97	1.50

LINEAR

78MG	1.75			LM710N	.79
LM106H	.99			LM711N	.39
LM300H	.80	LM340K-18	1.35	LM723N/H	.55
LM301CN/H	.35	LM340K-24	1.35	LM733N	1.00
LM302H	.75	LM340T-5	1.25	LM739N	1.19
LM304H	1.00	LM340T-6	1.25	LM741CN/H	.35
LM305H	.60	LM340T-8	1.25	LM741-14N	.39
LM307CN/H	.35	LM340T-12	1.25	LM747N/H	.79
LM308CN/H	1.00	LM340T-15	1.25	LM748N/H	.39
LM309H	1.10	LM340T-18	1.25	LM1310N	2.95
LM309K	1.25	LM340T-24	1.25	LM1458CN/H	.59
LM310CN	1.15	LM358N	1.00	MC1488N	1.95
LM311N/H	.90	LM370N	1.95	MC1489N	1.95
LM312H	1.95	LM373N	3.25	LM1496N	.95
LM317K	6.50	LM377N	4.00	LM1556V	1.75
LM318CN/H	1.50	LM380N	1.25	MC1741SCP	3.00
LM319N	1.30	LM380CN	.99	LM2111N	1.95
LM320K-5	1.35	LM381N	1.79	LM2901N	2.95
LM320K-5.2	1.35	LM382N	1.79	LM3053N	1.50
LM320K-12	1.35	NE501N	8.00	LM3065N	1.49
LM320K-15	1.35	NE510A	6.00	LM3900N(3401)	.49
LM320K-18	1.35	NE529A	4.95	LM3905N	.89
LM320K-24	1.35	NE531H/V	3.95	LM3909N	1.25
LM320T-5	1.25	NE536T	6.00	MC5558V	.59
LM320T-5.2	1.25	NE540L	6.00	8038B	4.95
LM320T-8	1.25	NE544N	4.95	LM75450N	.49
LM320T-12	1.25	NE550N	1.30	75451CN	.39
LM320T-15	1.25	NE555V	.39	75452CN	.39
LM320T-18	1.25	NE556N	.99	75453CN	.39
LM320T-24	1.25	NE560B	5.00	75454CN	.39
LM323K-5	5.95	NE561B	5.00	75491CN	.79
LM324N	1.80	NE562B	5.00	75492CN	.89
LM339N	.99	NE565N/H	1.25	75493N	.89
LM340K-5	1.35	NE566CN	1.75	75494CN	.89
LM340K-6	1.35	NE567V/H	.99	RC4136	1.25
LM340K-8	1.35	NE570N	4.95	RC4151	5.95
LM340K-12	1.35	LM703CN/H	.69	RC4194	5.95
LM340K-15	1.35	LM709N/H	.29	RC4195	4.49

74LS00TTL

74LS00	.23			74LS138	.69
74LS01	.23			74LS139	.69
74LS02	.23	74LS47	.69	74LS151	.69
74LS03	.23	74LS51	.23	74LS155	.69
74LS04	.29	74LS54	.23	74LS157	.69
74LS05	.29	74LS55	.23	74LS160	.89
74LS08	.23	74LS73	.35	74LS161	.89
74LS09	.29	74LS74	.35	74LS162	.89
74LS10	.23	74LS75	.49	74LS163	.89
74LS11	.60	74LS76	.35	74LS164	.99
74LS13	.49	74LS78	.39	74LS175	.79
74LS14	.99	74LS83	.75	74LS181	2.49
74LS15	.29	74LS85	.99	74LS190	.89
74LS20	.23	74LS86	.35	74LS191	.89
74LS21	.29	74LS90	.49	74LS192	.89
74LS22	.29	74LS92	.59	74LS193	.89
74LS26	.29	74LS93	.59	74LS194	.89
74LS27	.29	74LS95	.79	74LS195	.89
74LS28	.29	74LS96	.89	74LS253	.79
74LS30	.23	74LS107	.35	74LS257	.69
74LS32	.29	74LS109	.35	74LS258	1.39
74LS37	.35	74LS112	.35	74LS260	.55
74LS40	.29	74LS123	.99	74LS279	.59
74LS42	.69	74LS125	.69	74LS367	.59
		74LS132	.79	74LS368	.59
		74LS136	.39	74LS670	1.95

Vigilite

Electronic Home Security Timer

A microprocessor-based pre-programmed light control that fits into a home wall socket, replacing the standard on/off light switch. Discourages intruders and burglars by turning lights on and off in a "real-life" pattern while you're away.

Unlike other electromechanical timers Vigilite can simulate the lighting patterns of five different rooms as selected by the user. Vigilite can also control over-head lights, which other timers cannot. Three Vigilite units, simulating kitchen, bathroom, and bedroom lighting can give a home ultimate protection, because the user chooses a lighting pattern that depicts his real life pattern. He then sets the Vigilite clock and room pattern accordingly. (See bar chart below.) The house actually looks occupied, although no one is home.

Easy to install, the Vigilite unit contains an accurate digital LED clock, plus pre-programmed independent lighting patterns for bedroom, bathroom, kitchen, living room, and outside porch lights. All-solid-state components assure the user long product life and reliability.

Technical Specifications

1) Electrical requirement—120 VAC, 60Hz, 2½ Amps.
2) For use with permanently installed incandescent 40 W to 300 W light fixtures—single pole circuits.
3) Cannot be used to control television, appliances, or other motor driven equipment.
4) One vigilite unit required for each room circuit.

Part Number VGL-1

$39.95 ea.

DISCRETE LEDS

.200" dia.			**.125" dia.**		
XC556R	red	5/$1	XC209R	red	5/$1
XC556G	green	4/$1	XC209G	green	4/$1
XC556Y	yellow	4/$1	XC209Y	yellow	4/$1
XC556C	clear	4/$1			
.200" dia.			**.185" dia.**		
XC22R	red	5/$1	XC526R	red	5/$1
XC22G	green	4/$1	XC526G	green	4/$1
XC22Y	yellow	4/$1	XC526Y	yellow	4/$1
.170" dia.			XC526C	clear	4/$1
MV10B	red	4/$1	**.190" dia.**		
.085" dia.			XC111R	red	5/$1
MV50	red	6/$1	XC111G	green	4/$1
INFRA-RED LED 1/4"x1/4"x1/16" flat		5/$1	XC111Y	yellow	4/$1
			XC111C	clear	4/$1

TIMEX T1001
LIQUID CRYSTAL DISPLAY

FIELD EFFECT

4 DIGIT – .5" CHARACTERS
THREE ENUNCIATORS
2.00" X 1.20" PACKAGE
INCLUDES CONNECTOR

T1001-Transmissive	$7.95
T1001A-Reflective	8.25

DISPLAY LEDS

TYPE	POLARITY	HT	PRICE	TYPE	POLARITY	HT	PRICE
MAN 1	Common Anode-red	.270	2.95	MAN 6730	Common Anode-red ± 1	.560	.99
MAN 2	5 x 7 Dot Matrix-red	.300	4.95	MAN 6740	Common Cathode-red-D.D.	.560	.99
MAN 3	Common Cathode-red	.125	.25	MAN 6750	Common Cathode-red ± 1	.560	.99
MAN 4	Common Cathode-red	.187	1.95	MAN 6760	Common Anode-red	.560	.99
MAN 7G	Common Anode-green	.300	1.25	MAN 6780	Common Cathode-red	.560	.99
MAN 7Y	Common Anode-yellow	.300	.99	DL701	Common Anode-red ± 1	.300	.99
MAN 72	Common Anode-red	.300	.99	DL704	Common Cathode-red	.300	.99
MAN 74	Common Cathode-red	.300	1.25	DL707	Common Anode-red	.300	.99
MAN 82	Common Anode-yellow	.300	.99	DL728	Common Cathode-red	.500	1.49
MAN 84	Common Cathode-yellow	.300	.99	DL741	Common Anode-red	.600	1.25
MAN 3620	Common Anode-orange	.300	.99	DL746	Common Anode-red ± 1	.630	1.49
MAN 3630	Common Anode-orange ± 1	.300	.99	DL747	Common Anode-red	.600	1.49
MAN 3640	Common Cathode-orange	.300	.99	DL749	Common Cathode-red ± 1	.630	1.49
MAN 4610	Common Anode-orange	.300	.99	DL750	Common Cathode-red	.600	1.49
MAN 4640	Common Cathode-orange	.400	.99	DL338	Common Cathode-red	.110	.35
MAN 4710	Common Anode-red	.400	.99	FND70	Common Cathode	.250	.69
MAN 4730	Common Anode-red ± 1	.400	.99	FND358	Common Cathode ± 1	.357	.99
MAN 4740	Common Cathode-red	.400	.99	FND359	Common Cathode	.357	.75
MAN 4810	Common Anode-yellow	.400	.99	FND503	Common Cathode(FND500)	.500	.99
MAN 4840	Common Cathode-yellow	.400	.99	FND507	Common Anode (FND510)	.500	.99
MAN 6610	Common Anode-orange-D.D.	.560	.99	5082-7730	Common Anode-red	.300	1.30
MAN 6630	Common Anode-orange ± 1	.560	.99	HDSP-3400	Common Anode-red	.800	2.10
MAN 6640	Common Cathode-orange-D.D.	.560	.99	HDSP-3403	Common Cathode red	.800	2.10
MAN 6650	Common Cathode-orange ± 1	.560	.99	5082-7300	4 x 7 sgl. Digit-RHDP	.600	19.95
MAN 6660	Common Anode-orange	.560	.99	5082-7302	4 x 7 Sgl. Digit-LHDP	.600	19.95
MAN 6680	Common Cathode-orange	.560	.99	5082-7304	Overrange character (±1)	.600	15.00
MAN 6710	Common Anode-red-D.D.	.560	.99	5082-7340	4 x 7 Sgl. Digit-Hexadecimal	.600	22.50

RCA LINEAR

CA3013T	2.15	CA3082N	2.00
CA2023T	2.56	CA3083N	1.60
CA3035T	2.48	CA3086N	.85
CA3039T	1.35	CA3089N	3.75
CA3046N	1.30	CA3130T	1.39
CA3059N	3.25	CA3140T	1.25
CA3060N	3.25	CA3160T	1.25
CA3080T	.85	CA3401N	.49
CA3081N	2.00	CA3600N	3.50

CALCULATOR CHIPS/DRIVERS

MM5725	$2.95
MM5738	2.95
DM8864	2.00
DM8865	1.00
DM8887	.75
DM8889	.75
9374 7 seg. C.A. LED driver	1.50

CLOCK CHIPS

MM5309	$4.95
MM5311	4.95
MM5312	4.95
MM5314	4.95
MM5316	6.95
MM5318	9.95
MM5369	2.95
MM5387/1998A	4.95
MM5841	9.95

MOTOROLA

MC1408L7	$4.95
MC1408L8	5.75
MC1439L	2.95
MC3022P	2.95
MC3061P	3.50
MC4016(74416)	7.50
MC4024P	3.95
MC4040P	6.95
MC4044P	4.50

IC SOLDERTAIL — LOW PROFILE (TIN) SOCKETS

	1-24	25-49	50-100		1-24	25-49	50-100
8 pin LP	.$17	.16	.15	22 pin LP	$ .37	.36	.35
14 pin LP	.20	.19	.18	24 pin LP	.38	.37	.36
16 pin LP	.22	.21	.20	28 pin LP	.45	.44	.43
18 pin LP	.29	.28	.27	36 pin LP	.60	.59	.58
20 pin LP	.34	.32	.30	40 pin LP	.63	.62	.61

SOLDERTAIL STANDARD (TIN)

14 pin ST	$.27	.25	.24	28 pin ST	$ .99	.90	.81
16 pin ST	.30	.27	.25	36 pin ST	1.39	1.26	1.15
18 pin ST	.35	.32	.30	40 pin ST	1.59	1.45	1.30
24 pin ST	.49	.45	.42				

SOLDERTAIL STANDARD (GOLD)

8 pin SG	$.30	.27	.24	24 pin SG	$ .70	.63	.57
14 pin SG	.35	.32	.29	28 pin SG	1.10	1.00	.90
16 pin SG	.38	.35	.32	36 pin SG	1.65	1.40	1.26
18 pin SG	.52	.47	.43	40 pin SG	1.75	1.59	1.45

WIRE WRAP SOCKETS (GOLD) LEVEL #3

8 pin WW	$.39	.35	.31	22 pin WW	$ .95	.85	.75
10 pin WW	.45	.41	.37	24 pin WW	1.05	.95	.85
14 pin WW	.39	.38	.37	28 pin WW	1.40	1.25	1.10
16 pin WW	.43	.42	.41	36 pin WW	1.59	1.45	1.30
18 pin WW	.75	.68	.62	40 pin WW	1.75	1.55	1.40

1/4 WATT RESISTOR ASSORTMENTS – 5%

ASST. 1	5 ea.	10 OHM / 27 OHM	12 OHM / 33 OHM	15 OHM / 39 OHM	18 OHM / 47 OHM	22 OHM / 56 OHM	50 PCS	$1.75
ASST. 2	5 ea.	68 OHM / 180 OHM	82 OHM / 220 OHM	100 OHM / 270 OHM	120 OHM / 330 OHM	150 OHM / 390 OHM	50 PCS	1.75
ASST. 3	5 ea.	470 OHM / 1.2K	560 OHM / 1.5K	680 OHM / 1.8K	820 OHM / 2.2K	1K / 2.7K	50 PCS	1.75
ASST. 4	5 ea.	3.3K / 8.2K	3.9K / 10K	4.7K / 12K	5.6K / 15K	6.8K / 18K	50 PCS	1.75
ASST. 5	5 ea.	22K / 56K	27K / 68K	33K / 82K	39K / 100K	47K / 120K	50 PCS	1.75
ASST. 6	5 ea.	150K / 390K	180K / 470K	220K / 560K	270K / 680K	330K / 820K	50 PCS	1.75
ASST. 7	5 ea.	1M / 2.7M	1.2M / 3.3M	1.5M / 3.9M	1.8M / 4.7M	2.2M / 5.6M	50 PCS	1.75

ASST. 8R Includes Resistor Assortments 1-7 (350 PCS.) **$9.95 ea.**

$5.00 Minimum Order – U.S. Funds Only
California Residents – Add 6% Sales Tax

Spec Sheets – 25¢
1979 Catalog Available—Send 41¢ stamp

NEW 1979 Catalog

Jameco ELECTRONICS

a division of JAMES ELECTRONICS of California

PHONE ORDERS WELCOME (415) 592-8097

MAIL ORDER ELECTRONICS – WORLDWIDE
1021 HOWARD AVENUE, SAN CARLOS, CA 94070
Advertised Prices Good Thru January

TELEPHONE/KEYBOARD CHIPS

AY-5-9100	Push Button Telephone Dialler	$14.95
AY-5-9200	Repertory Dialler	14.95
AY-5-9500	CMOS Clock Generator	4.95
AY-5-2376	Keyboard Encoder (88 keys)	14.95
HD0165	Keyboard Encoder (16 keys)	7.95
74C922	Keyboard Encoder (16 keys)	9.95

ICM CHIPS

ICM7045	CMOS Precision Timer	24.95
ICM7205	CMOS LED Stopwatch/Timer	19.95
ICM7207	Oscillator Controller	7.50
ICM7208	Seven Decade Counter	19.95
ICM7209	Clock Generator	6.95

NMOS READ ONLY MEMORIES

MCM6571	128 X 9 X 7 ASCII Shifted with Greek	13.50
MCM6574	128 X 9 X 7 Math Symbol & Pictures	13.50
MCM6575	128 X 9 X 7 Alphanumeric Control Character Generator	13.50

MISCELLANEOUS

TL074CN	Quad Low Noise bi-fet Op Amp	2.49
TL494CN	Switching Regulator	4.49
TL496CP	Single Switching Regulator	1.75
11C90	Divide 10/11 Prescaler	19.95
95H90	Hi-Speed Divide 10/11 Prescaler	11.95
4N33	Photo-Darlington Opto-Isolator	3.95
MK50240	Top Octave Freq. Generator	17.50
DS0026CH	5Mhz 2-phase MOS clock driver	3.75
TIL308	.27" red num. display w/integ. logic chip	10.50
MM5320	TV Camera Sync. Generator	14.95
MM5330	4½ Digit DPM Logic Block	5.95
LD110/111	3½ Digit A/D Converter Set	25.00/set

LITRONIX ISO-LIT 1
Photo Transistor Opto-Isolator
(Same as MCT 2 or 4N25)
2/99¢

SN 76477
SOUND GENERATOR
Generates Complex Sounds
Low Power - Programmable
3.95 each

TV GAME CHIP AND CRYSTAL
AY-3-8500-1 and 2.01 MHZ Crystal (Chip & Crystal includes score display, 6 games and select angles, etc. **7.95/set**

EXAR

XR205	$3.40			XR2242CP	1.50
XR210	4.40			XR2264	4.25
XR215	4.40			XR2556	3.20
XR320	1.55	JE2206KA	14.95	XR2567	2.99
XR-L555	1.50	JE2206KB	19.95	XR3403	1.25
XR555	.39	XR1800	3.20	XR4136	1.25
XR556	.99	XR2206	4.40	XR4151	2.85
XR567CP	.99	XR2207	3.85	XR4194	1.45
XR567CT	1.25	XR2208	5.20	XR4202	3.60
XR1310P	1.30	XR2209	1.75	XR4212	2.05
XR1468CN	3.85	XR2211	5.25	XR4558	.75
XR1488	1.39	XR2212	4.35	XR4739	1.15
XR1489	1.39	XR2240	3.45	XR4741	1.47

DIODES

TYPE	VOLTS	W	PRICE
1N746	3.3	400m	4/1.00
1N751	5.1	400m	4/1.00
1N752	5.6	400m	4/1.00
1N753	6.2	400m	4/1.00
1N754	6.8	400m	4/1.00
1N757	9.0	400m	4/1.00
1N759	12.0	400m	4/1.00
1N959	8.2	400m	4/1.00
1N965	15	400m	4/1.00
1N5232	5.6	500m	28
1N5234	6.2	500m	28
1N5235	6.8	500m	28
1N5236	7.5	500m	28
1N5242	12	500m	28
1N5245	15	500m	28
1N456	25	40m	6/1.00
1N458	150	7m	6/1.00
1N485A	180	10m	5/1.00
1N4001	50 PIV 1 AMP		12/1.00

TYPE	VOLTS	W	PRICE
1N4002	100 PIV 1 AMP		12/1.00
1N4003	200 PIV 1 AMP		12/1.00
1N4004	400 PIV 1 AMP		12/1.00
1N4005	600 PIV 1 AMP		10/1.00
1N4006	800 PIV 1 AMP		10/1.00
1N4007	1000 PIV 1 AMP		10/1.00
1N3600	50	200m	6/1.00
1N4148	75	10m	15/1.00
1N4154	35	10m	12/1.00
1N4305	75	25m	15/1.00
1N4734	5.6	1w	28
1N4735	6.2	1w	28
1N4736	6.8	1w	28
1N4738	8.2	1w	28
1N4742	12	1w	28
1N4744	15	1w	28
1N1183	50 PIV 35 AMP		1.60
1N1184	100 PIV 35 AMP		1.70
1N1185	150 PIV 35 AMP		1.70
1N1186	200 PIV 35 AMP		1.80
1N1188	400 PIV 35 AMP		3.00

SCR AND FW BRIDGE RECTIFIERS

C36D	15A @ 400V	SCR(2N1849)	$1.95
C38M	35A @ 600V	SCR	1.95
2N2328	1.6A @ 300V	SCR	.50
MDA 980-1	12A @ 50V	FW BRIDGE REC.	1.95
MDA 980-3	12A @ 200V	FW BRIDGE REC.	1.95

TRANSISTORS

C106B1	.50			2N3904	4/1.00
MPSA05	.30	2N3055	.89	2N3905	4/1.00
MPSA06	5/1.00	MJE3055	1.00	2N3906	4/1.00
TIS97	6/1.00	2N3392	5/1.00	2N4013	3/1.00
TIS98	6/1.00	2N3398	5/1.00	2N4123	6/1.00
40409	1.75	PN3567	3/1.00	PN4249	4/1.00
40410	1.75	PN3568	4/1.00	PN4250	4/1.00
40673	1.75	PN3569	4/1.00	2N4400	4/1.00
2N918	4/1.00	MPS3638A	5/1.00	2N4401	4/1.00
2N2219A	2/1.00	MPS3702	5/1.00	2N4402	4/1.00
2N2221A	4/1.00	2N3704	5/1.00	2N4403	4/1.00
2N2222A	5/1.00	MPS3704	5/1.00	2N4409	5/1.00
PN2222 Plastic	7/1.00	2N3705	5/1.00	2N5086	4/1.00
2N2369	5/1.00	MPS3705	5/1.00	2N5087	4/1.00
2N2369A	4/1.00	2N3706	5/1.00	2N5088	4/1.00
MPS2369	5/1.00	MPS3706	5/1.00	2N5089	4/1.00
2N2484	4/1.00	2N3707	5/1.00	2N5129	5/1.00
2N2906	4/1.00	2N3711	5/1.00	PN5134	5/1.00
2N2907	5/1.00	2N3724A	.65	PN5138	5/1.00
PN2907 Plastic	7/1.00	2N3725A	1.00	2N5139	5/1.00
2N2925	5/1.00	2N3772	2.25	2N5210	5/1.00
MJE2955	1.25	2N3823	1.00	2N5449	3/1.00
2N3053	2/1.00	2N3903	5/1.00	2N5951	3/1.00

CAPACITOR CORNER

50 VOLT CERAMIC DISC CAPACITORS

	1-9	10-99	100		1-9	10-99	100 ·
10 pf	.05	.04	.03	.001µF	.05	.04	.035
22 pf	.05	.04	.03	.0047µF	.05	.04	.035
47 pf	.05	.04	.03	.01µF	.05	.04	.035
100 pf	.05	.04	.03	.022µF	.06	.05	.04
220 pf	.05	.04	.03	.047µF	.06	.05	.04
470 pf	.05	.04	.035	.1µF	.12	.09	.075

100 VOLT MYLAR FILM CAPACITORS

.001mf	.12	.10	.07	.022mf	.13	.11	.08
.0022	.12	.10	.07	.047mf	.21	.17	.13
.0047mf	.12	.10	.07	.1mf	.27	.23	.17
.01mf	.12	.10	.07	.22mf	.33	.27	.22

+20% DIPPED TANTALUMS (SOLID) CAPACITORS

.1/35V	.28	.23	.17	1.5/35V	.30	.26	.21
.15/35V	.28	.23	.17	2.2/25V	.31	.27	.22
.22/35V	.28	.23	.17	3.3/25V	.31	.27	.22
.33/35V	.28	.23	.17	4.7/25V	.32	.28	.23
.47/35V	.28	.23	.17	6.8/25V	.36	.31	.25
.68/35V	.28	.23	.17	10/25V	.40	.35	.29
1.0/35V	.28	.23	.17	15/25V	.63	.50	.40

MINIATURE ALUMINUM ELECTROLYTIC CAPACITORS

Axial Lead				Radial Lead			
.47/50V	.15	.13	.10	.47/25V	.15	.13	.10
1.0/50V	.16	.14	.11	.47/50V	.16	.14	.11
3.3/50V	.14	.12	.09	1.0/16V	.15	.13	.10
4.7/25V	.15	.13	.10	1.0/25V	.16	.14	.11
10/25V	.15	.13	.10	1.0/50V	.16	.14	.11
10/50V	.16	.14	.12	4.7/16V	.15	.13	.10
22/25V	.17	.15	.12	4.7/25V	.15	.13	.10
22/50V	.24	.20	.18	4.7/50V	.16	.14	.11
47/25V	.19	.17	.15	10/16V	.14	.12	.09
47/50V	.25	.21	.19	10/25V	.15	.13	.10
100/25V	.24	.20	.18	10/50V	.16	.14	.12
100/50V	.35	.30	.28	47/50V	.24	.21	.19
220/25V	.32	.28	.25	100/16V	.19	.15	.14
220/50V	.45	.41	.38	100/25V	.24	.20	.18
470/25V	.33	.29	.27	100/50V	.35	.30	.28
1000/16V	.55	.50	.45	220/16V	.23	.17	.16
2200/16V	.70	.62	.55	470/25V	.31	.28	.26

Circle 200 on inquiry card.



New Kits

Regulated Power Supply

- Uses LM 309K
- Heat sink provided
- P.C. board construction
- Provides a solid 1 amp @ 5V
- Includes components, hardware and instructions
- Sizes: 3-1/2" x 5" x 2" high

JE200 $14.95

Function Generator Kit



- Provides 3 basic waveforms: sine, triangle & square wave
- Frequency range from 1 Hz to 100K Hz
- Output amplitude from 0-volts to over 6 volts (peak to peak)
- Uses a 12V supply or a ± 6V split supply
- Includes chip, P.C. board, components and instructions.

JE2206B $19.95

Digital Stopwatch Kit



- Use Intersil 7205 Chip
- Plated thru double-sided P.C. Board
- LED display (red)
- Times to 59 min. 59.59 sec. with auto reset
- Quartz crystal controlled
- Three stopwatches in one: single event, split (cummulative) and taylor (sequential timing)
- Uses 3 penlite batteries
- Size: 4.5" x 2.15" x .90"

JE900 $39.95

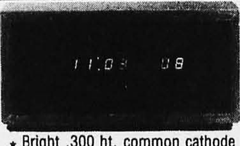
4-Digit Clock Kit



- Bright .357" ht. red display
- Sequential flashing colon
- 12 or 24 hour operation
- Extruded aluminum case (black)
- Pressure switches for hours, minutes and hold modes
- Includes all components, case and wall transformer
- Size: 3-1/4" x 1-3/4" x 1-1/4"

JE730 $14.95

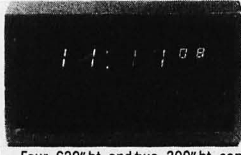
6-Digit Clock Kit



- Bright .300 ht. common cathode display
- Uses MM5314 clock chip
- Switches for hours, minutes and hold functions
- Hours easily viewable to 20 feet
- Simulated walnut case
- 115 VAC operation
- 12 or 24 hour operation
- Includes all components, case and wall transformer
- Size: 6-3/4" x 3-1/8" x 1-3/4"

JE701 $19.95

Jumbo 6-Digit Clock Kit



- Four .630" ht. and two .300" ht. common anode displays
- Uses MM5314 clock chip
- Switches for hours, minutes and hold functions
- Hours easily viewable to 30 feet
- Simulated walnut case
- 115 VAC operation
- 12 or 24 hour operation
- Includes all components, case and wall transformer
- Size: 6-3/4" x 3-1/8" x 1-3/4"

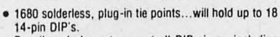
JE747 $29.95

MICROPROCESSOR COMPONENTS

8080A/8080A SUPPORT DEVICES		
8080A	CPU	$ 9.95
8212	8-Bit Input/Output	3.25
8214	Priority Interrupt Control	5.95
8216	Bi-Directional Bus Driver	3.49
8224	Clock Generator/Driver	3.95
8226	Bus Driver	3.49
8228	System Controller/Bus Driver	5.95
8238	System Controller	5.95
8251	Prog. Comm. 1/0 (USART)	7.95
8253	Prog. Interval Timer	14.95
8255	Prog. Periph. 1/0 (PPI)	9.95
8257	Prog. DMA Control	19.95
8259	Prog. Interrupt Control	19.95
6800/6800 SUPPORT DEVICES		
MC6800	MPU	$14.95
MC6802CP	MPU with Clock and Ram	24.95
MC6810API	128X8 Static Ram	5.95
MC6821	Periph. Inter. Adapt (MC6820)	7.49
MC6828	Priority Interrupt Conlroller	12.95
MC6830L8	1024X8 Bit ROM (MC68A30-8)	14.95
MC6850	Asynchronous Comm. Adapter	7.95
MC6852	Synchronous Serial Data Adapt.	9.95
MC6860	0-600 bps Digital MODEM	12.95
MC6862	2400 bps Modulator	14.95
MC6880A	Quad 3-State Bus. Trans. (MC8T26)	2.25
MICROPROCESSOR CHIPS—MISCELLANEOUS		
Z80(780C)	CPU	$19.95
Z80A(780-1)	CPU	24.95
CDP1802	CPU	19.95
2650	MPU	19.95
8035	8-Bit MPU w/clock, RAM, 1/0 lines	19.95
P8085	CPU	19.95
TMS9900JL	16-Bit MPU w/hardware, multiply & divide	49.95
SHIFT REGISTERS		
MM500H	Dual 25 Bit Dynamic	$.50
MM503H	Dual 50 Bit Dynamic	.50
MM504H	Dual 16 Bit Static	.50
MM506H	Dual 100 Bit Static	.50
MM510H	Dual 64 Bit Accumulator	.50
MM5016H	500/512 Bit Dynamic	.89
2504T	1024 Dynamic	3.95
2518	Hex 32 Bit Static	4.95
2522	Dual 132 Bit Static	2.95
2524	512 Static	.99
2525	1024 Dynamic	2.95
2527	Dual 256 Bit Static	2.95
2528	Dual 250 Static	4.00
2529	Dual 240 Bit Static	4.00
2532	Quad 80 Bit Static	2.95
2533	1024 Static	2.95
3341	Fifo	6.95
74LS670	4X4 Register File (TriState)	1.95
UART'S		
A-Y-5-1013	30K BAUD	5.95

MICROPROCESSOR MANUALS		
M-Z80	User Manual	$7.50
M-CDP1802	User Manual	7.50
M-2650	User Manual	5.00
ROM'S		
2513(2140)	Character Generator(upper case)	$9.95
2513(3021)	Character Generator(lower case)	9.95
2516	Character Generator	10.95
MM5230N	2048-Bit Read Only Memory	1.95

RAM'S			
1101	256X1	Static	$1.49
1103	1024X1	Dynamic	.99
2101(8101)	256X4	Static	3.95
2102	1024X1	Static	1.75
21L02	1024X1	Static	1.95
2111(8111)	256X4	Static	3.95
2112	256X4	Static MOS	4.95
2114	1024X4	Static 450ns	9.95
2114L	1024X4	Static 450ns low power	10.95
2114-3	1024X4	Static 300ns	10.95
2114L-3	1024X4	Static 300ns low power	11.95
5101	256X4	Static	7.95
5280/2107	4096X1	Dynamic	4.95
7489	16X4	Static	1.75
74S200	256X1	Static Tristate	4.95
93421	256X1	Static	2.95
UPD414 (MK4027)	4K	Dynamic 16 pin	4.95
UPD416 (MK4116)	16K	Dynamic 16 pin	14.95
TMS4044-45NL	4K	Static	14.95
TMS4045	1024X4	Static	14.95
2117	16,384X1	Dynamic 350ns (house marked)	9.95
MM5262	2KX1	Dynamic	4/1.00
PROM'S			
1702A	2048	FAMOS	$5.95
TMS2516 (2716)	16K*	EPROM(Intel 2716)	49.95
	*Requires single +5V power supply		
TMS2532	4KX8	EPROM	89.95
2708	8K	EPROM	10.95
2716 T.I	16K**	EPROM	29.95
	**Requires 3 voltages, —5V, +5V, +12V		
5203	2048	FAMOS	14.95
6301-1(7611)	1024	Tristate Bipolar	3.49
6330-1(7602)	256	Open C Bipolar	2.95
82S23	32X8	Open Collector	3.95
82S115	4096	Bipolar	19.95
82S123	32X8	Tristate	3.95
74186	512	TTL Open Collector	9.95
74188	256	TTL Open Collector	3.95
74S287	1024	Static	2.95



NEW!! IN STOCK... POWERACE

ALL-CIRCUIT EVALUATORS WITH POWER

- 1680 solderless, plug-in tie points...will hold up to 18 14-pin DIP's.
- Breadboard elements accept all DIP sizes...including RTL, DTL, TTL and CMOS devices. TO-5's and discretes with leads up to .032" dia.
- All connections to/from switches, indicators, power supplies and meters are made via solderless, plug-in, tie-point blocks on control panels.
- Interconnect with any solid 20 to 30 AWG wire.
- Breadboard elements are mounted on ground planes...ideal for high-frequency and high-speed/low-noise circuits.
- Short-circuit-proof fused power supplies.
- Operate on 110 to 130 VAC at 60 Hz
- Space-age compact styling and high-grade components permit convenient, organized and quick prototyping.
- All models are 7.5" wide 11.5" deep and 4.0" high (rear) 0.75" high (front) and weigh approx. 2.5 lbs.

POWERACE 101 -General purpose model for prototyping all types of circuit.
#923101 $ 84.95

POWERACE 102 -Complete digit prototyping lab with built-in logic probe
#923102 $114.95

POWERACE 103 -Triple-output power supply for prototyping both linear and digit circuits.
#923103 $124.95

B+K PRECISION 3½-Digit Portable DMM

- Overload Protected
- 3" high LED Display
- Battery or AC operation
- Auto Zeroing
- 1mv. 1Va. 0.1 ohm resolution
- Overange reading
- 10 meg input impendence
- DC Accuracy 1% typical

Ranges: DC Voltage - 0-1000V
AC Voltage 0-1000V
Freq Response 50-400 HZ
DC/AC Current: 0-100mA
Resistance 0-10 meg ohm
Size: 6.4" x 4.4" x 2"

Model 2800 $99.95
Comes with test leads, operating manual and spare fuse

Accessories:
AC Adapter BC-28 $9.00
Rechargeable Batteries BP-26 20.00
Carrying Case LC-28 7.50



100 MHz 8-Digit Counter

- 20 Hz-100 MHz Range
- .6" LED Display
- Crystal-controlled timebase
- Fully Automatic
- Portable — completely self-contained
- Size — 1.75" x 7.38" x 5.63"
- Four power souces, i.e. batteries, 110 or 220V with charger 12V with auto lighter adapter and external 7.2-10V power supply

MAX-100 $134.95



ACCESSORIES FOR MAX 100:
Mobile Charger Eliminator use power from car battery **Model 100 — CLA $3.95**
Charger/Eliminator use 110 V AC **Model 100 — CAI $9.95**

New CSC Mini-Max 6 Digit 50MHz Frequency Counter

- Guaranteed frequency range of 100 Hz to 50 MHz
- Full 6 digit display with antiglare window
- Fully automatic-range, polarity, slope, trigger, input level switching not required.
- Lead-zero blanking—All zeros to the left of the first non-zero digit are blanked. Kilo Hertz and Mega Hertz decimal points automatically light up when the unit is turned on.
- Built in input overvoltage protection.
- Use 9V Battery or 110/220V power.
- Complete with mini antenna.
- Lightweight — Only 8oz.

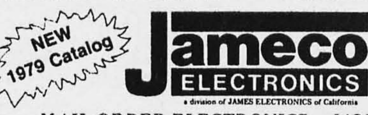
MINI-MAX $89.95

Accessories For Mini-Max

Part No.	Description	Price
MM-A4	Antenna	$ 3.95
MM-C5	Carrying case	5.95
MM-IPC	Input cable with clip leads	3.95
MM-AC2	110V adapter	9.95
MM-AC3	220V adapter	9.95

$5.00 Minimum Order – U.S. Funds Only
California Residents – Add 6% Sales Tax
Spec Sheets – 25¢
1979 Catalog Available—Send 41¢ stamp

NEW 1979 Catalog



PHONE ORDERS WELCOME (415) 592-8097

MAIL ORDER ELECTRONICS – WORLDWIDE
1021 HOWARD AVENUE, SAN CARLOS, CA 94070
Advertised Prices Good Thru January

The Incredible "Pennywhistle 103"



$139.95 Kit Only

The Pennywhistle 103 is capable of recording data to and from audio tape without critical speed requirements for the recorder and it is able to communicate directly with another modem and terminal for telephone "hamming" and communications. In addition, it is free of critical adjustments and is built with non-precision, readily available parts.

Data Transmission Method .....Frequency-Shift Keying, full-duplex (half-duplex selectable).
Maximum Data Rate..........300 Baud.
Data Format ....Asynchronous Serial (return to mark level required between each character).
Receive Channel Frequencies ...2025 Hz for space; 2225 Hz for mark.
Transmit Channel Frequencies ..Switch selectable: Low (normal) = 1070 space, 1270 mark; High = 025 space, 2225 mark.
Receive Sensitivity............−46 dbm accoustically coupled.
Transmit Level ....−15 dbm nominal. Adjustable from −6 dbm to −20 dbm.
Receive Frequency Tolerance ...Frequency reference automatically adjusts to allow for operation between 1800 Hz and 2400 Hz.
Digital Data Interface..........EIA RS-232C or 20 mA current loop (receiver is optoisolated and non-polar).
Power Requirements ....120 VAC, single phase, 10 Watts.
Physical......................All components mount on a single 5" by 9" printed circuit board. All components included.
Requires a VOM, Audio Oscillator Frequency Counter and/or Oscilloscope to align.

TRS-80 16K Conversion Kit

Expand your 4K TRS-80 System to 16K. Kit comes complete with:
* 8 each UPD416 (16K Dynamic Rams)
* Documentation for conversion

TRS-16K $115.00

Special Offer - Order both your TRS-16K and the Sup'R' MOD II Interface kit together (retail value $144.95) **for only $139.95**

COMPUTER CASSETTES

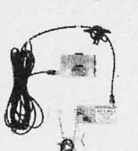


- 6 EACH 15 MINUTE HIGH QUALITY C-15 CASSETTES
- PLASTIC CASE INCLUDED 12 CASSETTE CAPACITY
- ADDITIONAL CASSETTES AVAILABLE #C-15-$2.50 ea

CAS-6 $14.95
(Case and 6 Cassettes)

SUP 'R' MOD II

UHF Channel 33 TV Interface Unit Kit



Wide Band B/W or Color System
- Converts TV to Video Display for home computers, CCTV camera, Apple II, works with Cromeco Dazzler, SOL-20, IRS-80, Challenger, etc.
- MOD II is pretuned to Channel 33 (UHF).
- Includes coaxial cable and antenna transformer.

MOD II $29.95 Kit

Custom Cables & Jumpers



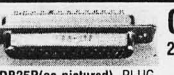
DB 25 Series Cables

Part No.	Cable Length	Connectors	Price
DB25P-4-P	4 Ft.	2-DP25P	$15.95 ea.
DB25P-4-S	4 Ft.	1-DP25P/1-25S	$16.95 ea.
DB25S-4-S	4 ft.	2-DP25S	$17.95 ea.
Dip Jumpers			
DJ14-1	1 ft.	1-14 Pin	$1.59 ea.
DJ16-1	1 ft.	1-16 Pin	1.79 ea.
DJ24-1	1 ft.	1-24 Pin	2.79 ea.
DJ14-1-14	1 ft.	2-14 Pin	2.79 ea.
DJ16-1-16	1 ft.	2-16 Pin	3.19 ea.
DJ24-1-16	1 ft.	2-24 Pin	4.95 ea.

For Custom Cables & Jumpers, See JAMECO 1979 Catalog for Pricing

CONNECTORS

25 Pin-D Subminiature



DB25P(as pictured)	PLUG	$2.95
DB25S	SOCKET	3.50
DB51226-1	Cable Cover for DB25 P or S	1.75

63-Key Unencoded Keyboard

This is a 63-key, terminal keyboard newly manufactured by a large computer manufacturer. It is unencoded with SPST keys, unattached to any kind of PC board. A very solid molded plastic 13 x 4" base suits most application. IN STOCK **$29.95/each**

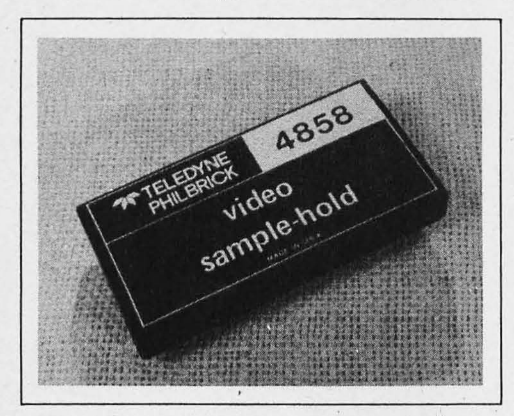
Hexadecimal Unencoded Keypad



19-key pad includes 1-10 keys, ABCDEF and 2 optional keys and a shift key. **$10.95/each**

Circle 200 on inquiry card.

Video Sample and Hold Amplifier Features Fast Acquisition Time



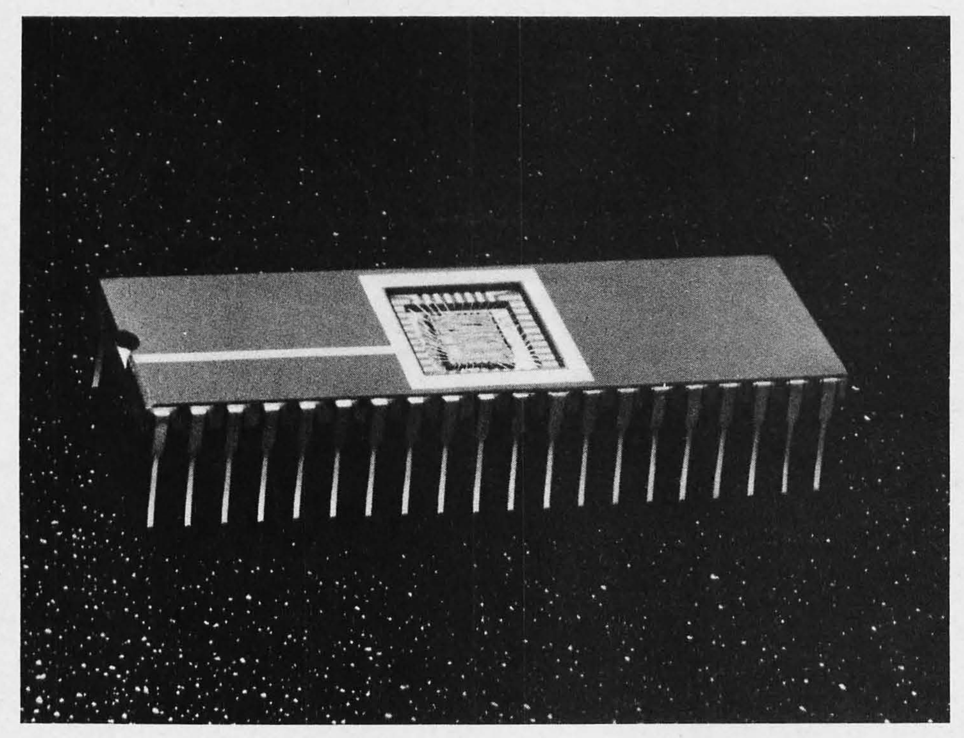
The 4858 video sample and hold amplifier features an acquisition time of 130 ns to 0.01% and permits conversion rates of up to 5 MHz for 12 bit analog to digital conversion. The unit acquires signals to 8 bit, or 0.2%, accuracies in 75 ns.

The 4858 is user programmable for either transistor-transistor logic or emitter coupled logic compatability. Applications include fast Fourier transforms, radar pulse analysis, and deglitching.

The unit is housed in a 4 by 2 by 0.6 inch (10.16 by 5.08 by 1.52 cm) metal case, providing electrostatic shielding. The 4858 is priced at $440 in unit quantities. For further information contact Teledyne Philbrick, Allied Dr at Rt 128, Dedham MA 02026.■

Circle 528 on inquiry card.

IBM Compatible Double Density Disk Formatter and Controller



An IBM compatible double density floppy disk formatter and controller integrated circuit, called the FD 1791, is now available from Western Digital Corp, 3128 Red Hill Av, Newport Beach CA 92663. It is designed to meet the requirements of the most widely used density mode (FM) and System 34 double density mode (MFM) applications.

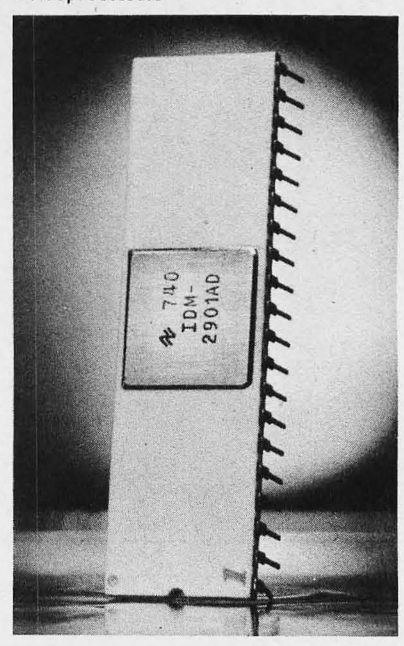
The FD 1791 interfaces a processor to a floppy disk drive. The device provides the data accessing controls and the bidirectional transfer of information between the processor memory and the magnetically stored data on the disk. The disk data is stored in a data entry format compatible with either single or double density modes.

The FD 1791 is able to read, write and format a double density disk by means of its address mark detection, FM and MFM encode and decode logic, window extension, and write precompensation. Another feature allows operation in two modes: one for floppy disk and another for minifloppy. The first uses a clock input of 2 MHz and provides stepping rates of 3, 6, 10 and 20 ms. The clock input for the second is set at 1 MHz and the stepping rates change to 6, 12, 20 and 40 ms. The clock inputs are not altered when changing between single and double density.

The FD 1791 is priced at $43.20 for quantities of 100.■

Circle 529 on inquiry card.

High Speed 4 Bit Slice 2900 Bipolar Microprocessors



Using a new advanced bipolar LSI process, National Semiconductor Corp, 2900 Semiconductor Dr, Santa Clara CA 95051, has developed a family of higher speed 2900 type 4 bit slice microprocessor components. According to the company, they are 30 to 50 percent faster than similar designs now on the market. Designated the IDM2900 family, the new devices, 16 in all, use a process that combines low power Schottky peripheral electronics with proprietary high speed Tri-State emitter coupled logic electronics for interface. National calls this new process "SCL." For complete information on the IDM2900 family contact the company.■

Circle 530 on inquiry card.

Two New Schottky Programmable Read Only Memories



Two new 4096 bit Schottky programmable read only memories, the 74S476 and the 74S477, have been added to Texas Instruments' bipolar memory product line. Typical access time is 35 ns. These programmable read only memories are offered in plastic (N suffix) and ceramic (J suffix) dual-in-line packages.

The devices can serve as direct replacements for any currently offered 18 pin 4 K byte memories. The prices in lots of 100 range from $9.30 to $18.60, depending on the package.

Inquiries should be forwarded to Texas Instruments Inc, Inquiry Answering Service, POB 5012, Dallas TX 75222.■

Circle 531 on inquiry card.

BECKIAN ENTERPRISES



All Prime Quality — New Parts Only

Satisfaction Guaranteed

EDGE CARD CONNECTORS: GOLD PLATED.

BODY: Non brittle, solvent resistant, high temp, G.E. Valox. The finest you can buy.
CONTACTS: Bifurcated Phos./Bronze; Gold/Nickel.

ALTAIR S-100: Cont./Ctrs. .125" Row Spacing, .140"

50/100 Dip Sold.	$3.95 ea.	5 pcs.	$3.75 ea.
50/100 Sold. Eye.	6.95 ea.	5 pcs.	6.50 ea.

IMSAI S-100: Cont./Ctrs. .125" Row Spacing, .250"

50/100 Dip Sold.	$4.20 ea.	5 pcs.	$3.95 ea.
50/100 W/Wrap 3	3.75 ea.	5 pcs.	3.50 ea.
IMSAI CARD GUIDES:	0.19 ea.	5 pcs.	0.16 ea.

CROMEMCO S-100: Cont./Ctrs. .125" Row Spacing, .250"

50/100 Dip Sold. (Or short W/Wrap)	$6.50 ea.	5 pcs.	$6.00 ea.

OTHER CONNECTORS AVAILABLE

.100" Contact Ctrs., .140" Row Spacing.

22/44 Dip Sold.	$2.30 ea.	5 pcs.	$2.10 ea.
25/50 Sold. Eye.	2.95 ea.	5 pcs.	2.75 ea.
40/80 Sold. Eye.	4.80 ea.	5 pcs.	4.50 ea.
43/86 Dip Sold.	4.90 ea.	5 pcs.	4.70 ea.
43/86 Sold. Eye.	4.90 ea.	5 pcs.	4.70 ea.

.156" Contact Ctrs., .140" Row Spacing.

6/– Sgle. Row (PET)	$1.00 ea.	5 pcs.	$0.90 ea.
22/44 Sold. Eye. (KIM)	1.90 ea.	5 pcs.	1.80 ea.
22/44 Dip Sold. (KIM)	1.90 ea.	5 pcs.	1.80 ea.
43/86 Dip Sold.	4.90 ea.	5 pcs.	4.70 ea.

.156" Contact Ctrs., .200" Row Spacing.

15/30 W/Wrap 3	$1.05 ea.	5 pcs.	$0.95 ea.
22/44 W/Wrap 3	2.30 ea.	5 pcs.	2.10 ea.
36/72 Sold. Eye.	3.45 ea.	5 pcs.	3.30 ea.
36/72 W/Wrap 3	3.85 ea.	5 pcs.	3.70 ea.
43/86 W/Wrap 3	5.50 ea.	5 pcs.	5.00 ea.

POLARIZING KEYS FOR ALL OF THE ABOVE:
Specify: IN Contact or BETWEEN Contact:
1 to 49 pcs. $0.10 ea. 50 pcs./Up $0.08 ea.

SPECIAL
12/24 Pin .156" Cont./Ctrs. .200" Row Spacing.
TIN PLATED CONTACTS.
IDEAL FOR PET INTERFACE & PARALLEL USER PORT.
$1.25 ea. 5 pcs. $1.10 ea.

SUBMINIATURE CONNECTORS: (DB 25 SERIES, RS 232.)

DB 25P Male Plug	$2.50 ea.	5 pcs.	$2.20 ea.
DB 25S Female Socket	3.60 ea.	5 pcs.	3.40 ea.
DB 51212-1 Grey Hood	1.20 ea.	5 pcs.	1.10 ea.
DB 51226-1A Black Hood	1.30 ea.	5 pcs.	1.20 ea.
D 20418-2 Hardware Set	0.75 ea.	5 pcs.	0.70 ea.

SAVE: BUY A SET: (1 DB25P, 1 DB25S, Any Hood.)
1 Set: $6.35 ea. 5 sets: $6.15 ea.
NOTE: For Hardware, (D20418-2) Add $.65/Set.

WHISPER FANS
Excellent for computer cabinet cooling. This is the most quiet fan you will find. Only measures 4 3/4" square by 1½" deep. U. L. Listed.
$21.00 ea. 5 pcs. $19.00 ea.

I. C. SOCKETS. GOLD. WIRE WRAP 3 TURN.
14 pin $0.36 ea.
16 pin 0.38 ea.

I. C. SOCKETS. Dip Solder. Tin.
14 pin $0.15 ea.
16 pin 0.17 ea.

2708 EPROMS PRIME
$14.00 ea.

8080 PRIME
$9.00 ea.

WRITE FOR LARGER QUANTITY DISCOUNTS. DEALER INQUIRIES ARE WELCOME.

WE ARE CONNECTOR (EDGE CARD) SPECIALISTS. IF YOU DO NOT SEE WHAT YOU NEED IN THIS ADVERTISEMENT, PLEASE WRITE US. WE WILL REPLY.

TERMS: Minimum Order $10.00: Add $1.25 for handling and shipping. All orders over $25.00 in USA and Canada: WE PAY THE SHIPPING.
NOTE: CA residents please add 6% sales tax.
NO C.O.D. SHIPMENTS OR ORDERS ACCEPTED.

MAIL ORDERS TO: *Beckian Enterprises*
P.O. Box 3089
Simi Valley, CA 93063

ATWOOD ENTERPRISES

KITS

$ 79.95	**4K RAM**	Available assembled and tested $89.95.
$129.95	**4K PROM**	Bipolar 512 x 8 Proms 93448/6341.
$149.95	**8K EPROM**	Needs only 4K space 2716.
$ 59.95	**DIGITAL I/O**	8 parallel ports plus 16 interrupts.
$ 99.95	**ANALOG IN**	32 inputs, 8 bits, 100 microseconds.
$ 49.95	**PROM PROGRAMMING**	Burns 4. Can copy to 3.
$ 30.00	**MOTHER BOARD**	8 slot with connectors and card guides.

DID YOU KNOW . . .

BIG SALE

1 each board:

Digital I/O
Eprom programming

8K Eprom Board (without 2716's)

REGULAR PRICE $139.85

JANUARY ONLY $100.00

Extra zero insertion force sockets $5.00

MOTHER BOARD FREE

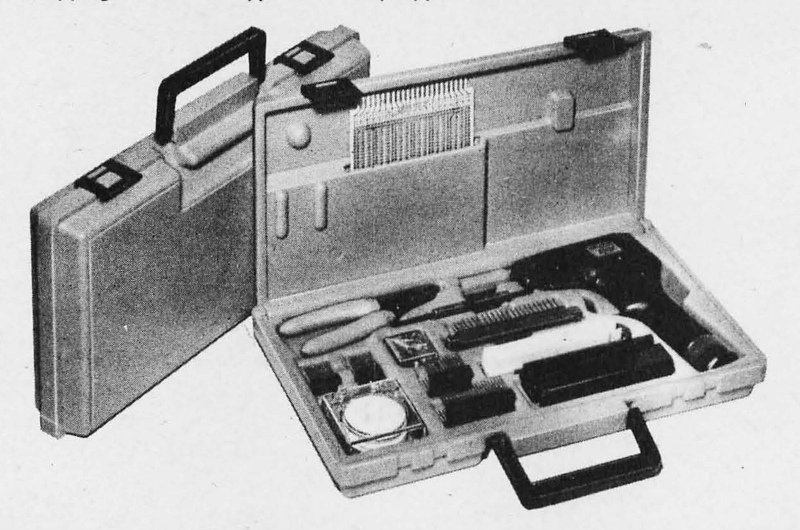
WITH PURCHASE OF
4 RAM BOARD KITS

MAKE CHECK OR MONEY ORDER PAYABLE TO:
Kathryn Atwood Enterprises
P.O. Box 5203, Orange, CA 92667

Discounts available at OEM quantities. For orders less than $25 total, add $1.25 for shipping. California residents add 6% sales tax. Estimated shipping time 2 days ARO with money order. For checks allow 7 days for check to clear.

What's New?

Wire Wrapping Kit for Prototype and Hobby Applications

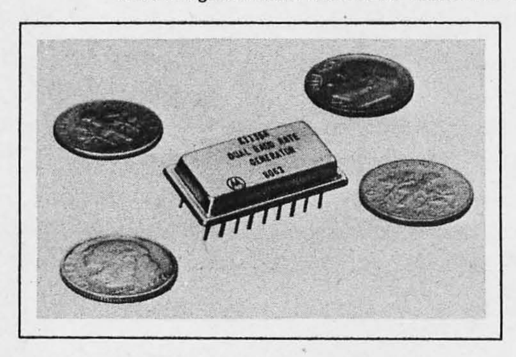


Model WK-5B is a new wire wrapping kit that contains a complete range of tools and parts for prototype and hobby applications. The kit includes: Model BW-630 battery wire wrapping tool complete with bit and sleeve; Model WSU-30, a hand wire wrapping, unwrapping and stripping tool; a universal printed circuit board; an edge connector with wire wrapping terminals; a set of printed circuit card guides and brackets; a minishear with safety clip; 14, 16, 24 and 40 pin dual-in-line package (DIP) sockets; an assortment of wire wrapping terminals; a DIP inserter and extractor; and a 3 color wire dispenser complete with 50 feet each of red, white and blue Kynar insulated silver plated solid AWG 30 copper wire.

The Model WK-5B wire wrapping kit comes in a durable plastic case and is priced at $74.95. For further information contact OK Machine and Tool Corp, 3455 Conner St, Bronx NY 10475.■

Circle 648 on inquiry card.

Dual Programmable Data Rate Generator from Motorola



This K1135A dual data rate generator is a self-contained quartz crystal in hybrid form. The chip and crystal are packaged in an 18 pin dual-in-line package 0.2 inches high.

The K1135A simultaneously generates a transmit and receive frequency at 16 times the transmission rate. 16 standard transmission rates can be generated by manual control or by a 4 bit address programmed in software.

Other features include a miniature size, 1.015 by 0.515 by 0.200 inches (2.578 by 1.308 by .508 cm), ±.01% stability, and a separate clock frequency output (5.0688 MHz). For further information, contact Motorola Component Products, 2553 N Edington, Franklin Park IL 60131.■

Circle 649 on inquiry card.

New Metric Heat Shrinkable Tubing

The ST221-M series of shrinkable all-purpose, irradiated polyolefin extruded tubing is suited for most electrical and electronic metric applications. The tubing is thermally stable and provides a tight mechanical bond.

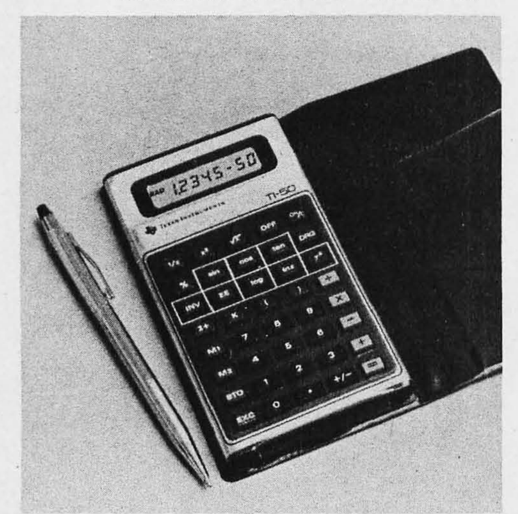
Tubing sizes range from 1.191 mm (0.047 inches) through 25.4 mm (1 inch) and are available in standard lengths of 1.219 meters (47.99 inches) in various standard packages of 6.096 meters (20 feet) and 30.48 meters (100 feet).

The metric tubing features a shrink ratio of 2:1 with only 5 percent longitudinal shrinkage.

Stock colors are black, white, red, yellow, blue and clear. Detailed information and complete specifications are available from the Cole-Flex Corp, 91 Cabot St, W Babylon NY 11704.■

Circle 651 on inquiry card.

The Slimline TI-50 Fits in Your Pocket



The Slimline TI-50 is a scientific pocket calculator with statistics, from Texas Instruments. The TI-50 offers 60 algebra calculator functions, including logarithms, trigonometry and statistics. It also includes two full arithmetic memories with a new constant memory feature which allows the calculator to retain data entered in the memories whether the unit is on or off. The constant memory enables retention of continuously used constants, values and statistics, reducing the time needed to perform repetitive entries and operations. The unit also features TI's algebraic operating system (AOS), which includes 15 levels of parentheses and up to four pending operations. The automatic power down (APD) automatically shuts the calculator off after 15 minutes of nonuse.

Other features include a large liquid crystal display (LCD) 8 digit readout, and scientific notation (5 digit mantissa, 2 digit exponent display format) with mantissa expansion capability. A special battery indicator provides a convenient check on battery condition.

The TI-50 is priced at $35. For further information contact Texas Instruments Inc, Consumer Relations, POB 53 (Attn: TI-50), Lubbock TX 79408.■

Circle 650 on inquiry card.

Diskette Filing System

The Super Diskette Filing System is an easel type vinyl covered binder with 20 specially designed 8½ by 11 inch vinyl 3 ring binder inserts. Each insert holds two disks and provides space for noting contents in an abbreviated form beside the disk and a larger pouch over the bottom of the disk for more extensive notes.

Diskettes need only be labeled with a small numbered label to associate them with their pouch. A separate pouch is provided for self-stick labels and write protect tabs.

Included with the package is a punch, template and complete instructions for modifying single side disks so that both surfaces can be used.

The retail price for this system is $69.95. Contact Aaron Associates, POB 1720A, Garden Grove CA 92640.■

Circle 652 on inquiry card.



COMPUTER INTERFACES & PERIPHERALS

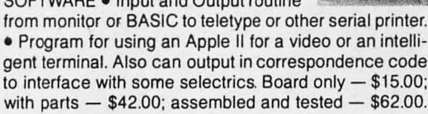
SAVE

For free catalog including parts lists and schematics, send a self-addressed stamped envelope.

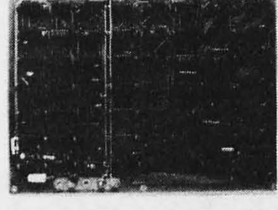


APPLE II SERIAL I/O INTERFACE *

Part no. 2
Baud rate is continuously adjustable from 0 to 30,000 • Plugs into any peripheral connector • Low current drain. RS-232 input and output • On board switch selectable 5 to 8 data bits, 1 or 2 stop bits, and parity or no parity either odd or even • Jumper selectable address • SOFTWARE • Input and Output routine from monitor or BASIC to teletype or other serial printer. • Program for using an Apple II for a video or an intelligent terminal. Also can output in correspondence code to interface with some selectrics. Board only — $15.00; with parts — $42.00; assembled and tested — $62.00.



T.V. TYPEWRITER

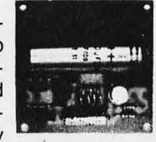


Part no. 106
• Stand alone TVT • 32 char/line, 16 lines, modifications for 64 char/line included • Parallel ASCII (TTL) input • Video output • 1K on board memory • Output for computer controlled curser • Auto scroll • Non-destructive curser • Curser inputs: up, down, left, right, home, EOL, EOS • Scroll up, down • Requires +5 volts at 1.5 amps, and -12 volts at 30 mA • All 7400, TTL chips • Char. gen. 2513 • Upper case only • Board only $39.00; with parts $145.00

8K STATIC RAM

Part no. 300
• 8K Altair bus memory • Uses 2102 Static memory chips • Memory protect • Gold contacts • Wait states • On board regulator • S-100 bus compatible • Vector input option • TRI state buffered • Board only $22.50; with parts $160.00

RF MODULATOR *



Part no. 107
• Converts video to AM modulated RF, Channels 2 or 3. So powerful almost no tuning is required. On board regulated power supply makes this extremely stable. Rated very highly in Doctor Dobbs' Journal. Recommended by Apple. • Power required is 12 volts AC C.T., or +5 volts DC • Board $7.60; with parts $13.50

MODEM *

Part no. 109
• Type 103 • Full or half duplex • Works up to 300 baud • Originate or Answer • No coils, only low cost components • TTL input and output-serial • Connect 8 ohm speaker and crystal mic. directly to board • Uses XR FSK demodulator • Requires +5 volts • Board $7.60; with parts $27.50

TIDMA *

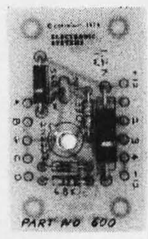
Part no. 112
• Tape Interface Direct Memory Access • Record and play programs without bootstrap loader (no prom) has FSK encoder/decoder for direct connections to low cost recorder at 1200 baud rate, and direct connections for inputs and outputs to a digital recorder at any baud rate. • S-100 bus compatible • Board only $35.00; with parts $110.00

DC POWER SUPPLY *



Part no. 6085
• Board supplies a regulated +5 volts at 3 amps., +12, -12, and -5 volts at 1 amp. • Power required is 8 volts AC at 3 amps., and 24 volts AC C.T. at 1.5 amps. • Board only $12.50; with parts excluding transformers $42.50

RS 232/TTY * INTERFACE

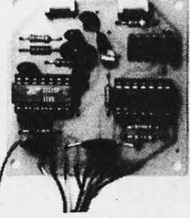


Part no. 600
• Converts RS-232 to 20mA current loop, and 20mA current loop to RS-232 • Two separate circuits • Requires +12 and -12 volts • Board only $4.50, with parts $7.00

TAPE INTERFACE *

Part no. 111
• Play and record Kansas City Standard tapes • Converts a low cost tape recorder to a digital recorder • Works up to 1200 baud • Digital in and out are TTL-serial • Output of board connects to mic. in of recorder • Earphone of recorder connects to input on board • No coils • Requires +5 volts, low power drain • Board $7.60; with parts $27.50

UART & BAUD RATE GENERATOR *



Part no. 101
• Converts serial to parallel and parallel to serial • Low cost on board baud rate generator • Baud rates: 110, 150, 300, 600, 1200, and 2400 • Low power drain +5 volts and -12 volts required • TTL compatible • All characters contain a start bit, 5 to 8 data bits, 1 or 2 stop bits, and either odd or even parity. • All connections go to a 44 pin gold plated edge connector • Board only $12.00; with parts $35.00 with connector add $3.00

RS 232/TTL* INTERFACE

Part no. 232
• Converts TTL to RS-232, and converts RS-232 to TTL • Two separate circuits • Requires -12 and +12 volts • All connections go to a 10 pin gold plated edge connector • Board only $4.50; with parts $7.00 with connector add $2.00



ELECTRONIC SYSTEMS Dept. B, P.O. Box 21638, San Jose, CA. USA 95151

To Order: Mention part number and description. For parts kits add "A" to part number. In USA, shipping paid for orders accompanied by check, money order, or Master Charge, BankAmericard, or VISA number, expiration date and signature. Shipping charges added to C.O.D. orders. California residents add 6.5% for tax. Outside USA add 10% for air mail postage, no C.O.D.'s. Checks and money orders must be payable in US dollars. Parts kits include sockets for all ICs, components, and circuit board. Documentation is included with all products. All items are in stock, and will be shipped the day order is received via first class mail. Prices are in US dollars. No open accounts. To eliminate tariff in Canada boxes are marked "Computer Parts." Dealer inquiries invited.
24 Hour Order Line: (408) 226-4064

✱ Circuits designed by John Bell





NEW IN STOCK

The EW-2001 A "Smart" VIDEO BOARD KIT At A "Dumb" Price!

A VIDEO BOARD + A MEMORY BOARD + AN I/O BOARD — ALL IN ONE!

■ STATE OF THE ART TECHNOLOGY USING DEDICATED MICROPROCESSOR I.C.
■ NUMBER OF I.C.s REDUCED BY 50% FOR HIGHER RELIABILITY ■ MASTER PIECE OF ENGINEERING ■ FULLY SOFTWARE CONTROLLED

Priced at ONLY **$199.95**
Basic Software Included

SPECIAL FEATURES:

- S-100 bus compatible
- Parallel keyboard port
- On board 4K screen memory (optional)* relocatable to main computer memory
- Text editing capabilities (software optional)
- Scrolling: up and down through video memory
- Blinking characters
- Reversed video
- Provision for on board ROM
- CRT and video controls fully programmable (European TV)
- Programmable no. of scan lines
- Underline blinking cursor
- Cursor controls: up, down, left, right, home, carriage return
- Composite video

*Min. 2K required for operation of this board.

DISPLAY FEATURES:

- 128 displayable ASCII characters (upper and lower case alphanumeric, controls)
- 64 or 32 characters per line (jumper selectable)
- 32 or 16 lines (jumper selectable)
- Screen capacity 2048 or 512
- Character generation: 7 x 11 dot matrix

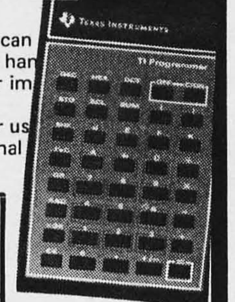
OPTIONS:

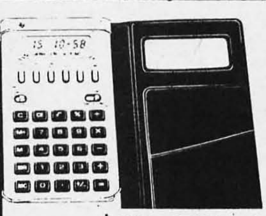
Sockets	$10.00
2K Static Memory (with Sockets)	$45.00
4K Static Memory (with Sockets)	$90.00
Complete unit, assembled and tested with 4K Memory	$335.00
Basic software on ROM	$20.00
Text editor on ROM	$75.00

DEALER INQUIRIES WELCOMED

TI PROGRAMMER $53.95

Hexadecimal. Octal. Decimal.
Enter a number in base 8, 10, or 16. TI Programmer can quickly convert to either of the other bases. Rapidly han arithmetic in all three bases giving you more time for im ant programming or troubleshooting tasks.
Ideal for use with any size computer. TI Programmer us integer "two's complement" arithmetic in hexadecimal octal bases.





$44.95 DATA CHROM

Large, easy-to-read 8-digit liquid crystal display.
Clock mode displays time, day, date, and AM/PM.
Stopwatch mode displays hours, minutes, seconds and tenths of seconds up to 9-59-59.9.
Economical—you'll get typically 12 months normal operation on a single set of batteries.
Attractive—comes in brown vinyl wallet folder with insert pockets for business cards. Makes a neat addition to your personal or business accessories.
24-hour alarm.
Stopwatch records and displays lap and total elapsed times. Up to one-tenth of a second accuracy.

SHIPPING: $3.50

California residents add 6% sales tax

ELECTRONICS WAREHOUSE Inc.

1603 AVIATION BLVD. Dept.
REDONDO BEACH, CA. 90278
TEL. (213) 376-8005

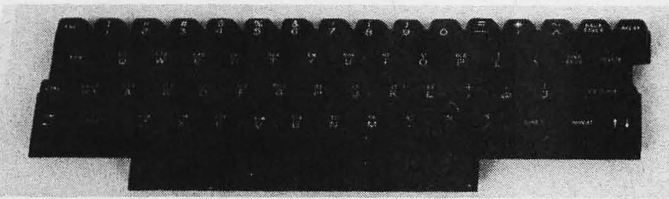


master charge

WRITE FOR FREE CATALOG

Minimum Order: $10

ASCII KEYBOARD KIT $74.00



Additional Improvements: Double Size Return Key Control Characters Molderd on Key Caps

- Power: +5V 275mA
- Upper and Lower Case
- Full ASCII Set
- 7 or 8 Bits Parallel Data
- Optional Serial Output
- Selectable Positve or Negative Strobe, and Strobe Pulse Width
- 2 Key Roll-Over
- 3 User DEfineable Keys
- P.C. Board Size: 17-3/16" x 5"

OPTIONS:

■ Metal Enclosure Painted Blue and White	$27.50
■ 18 Pin Edge Con.	$ 2.00
■ I.C. Sockets	$ 4.00
■ Serial Output Provision (Shift Register)	$ 2.00
■ Upper Case Lock Switch for Capital Letters and Nos.	$ 2.00
Assembled (on Sockets) and Tested	$90.00

APPLE II I/O BOARD KIT

Plugs Into Slot of Apple II Mother Board

18 Bit Parallel Output Port (Expandable to 3 Ports)
1 Input Port
15mA Output Current Sink or Source
Can be used for peripheral equipment such as printers, floppy discs, cassettes, paper tapes, etc.

1 free software listing for SWTP PR40 or IBM selectric

PRICE:
1 Input and 1 Output Port for **$49.00**
1 Input and 3 Output Ports for **$64.00**
Dealer Inquiries Invited

What's New?

SOFTWARE

COMPAS Reduces Prices on 6500 Processor Cross Assemblers

COMPAS has reduced the prices on all its cross assemblers for the 6500 processor family. The 6500 is built by MOS Technology, SYNERTEK and Rockwell. All cross assemblers are now priced at $600 which includes test programs and one year free support. Two versions of the cross assemblers are offered.

The MINmic 1165 cross assembler is available for any PDP-11 using the RT-11 operating system and was formerly priced at $900. The MINmic cross assembler is written in MACRO 11 for maximum speed and requires 5 K words memory. Normal distribution is on floppy disk or RK05 disk. This assembler may be used in conjunction with the CSL/65 cross compiler which also runs on any similar PDP-11.

For further information contact COMPAS, Computer Applications Corp, 413 Kellogg, Ames IA 50010.■

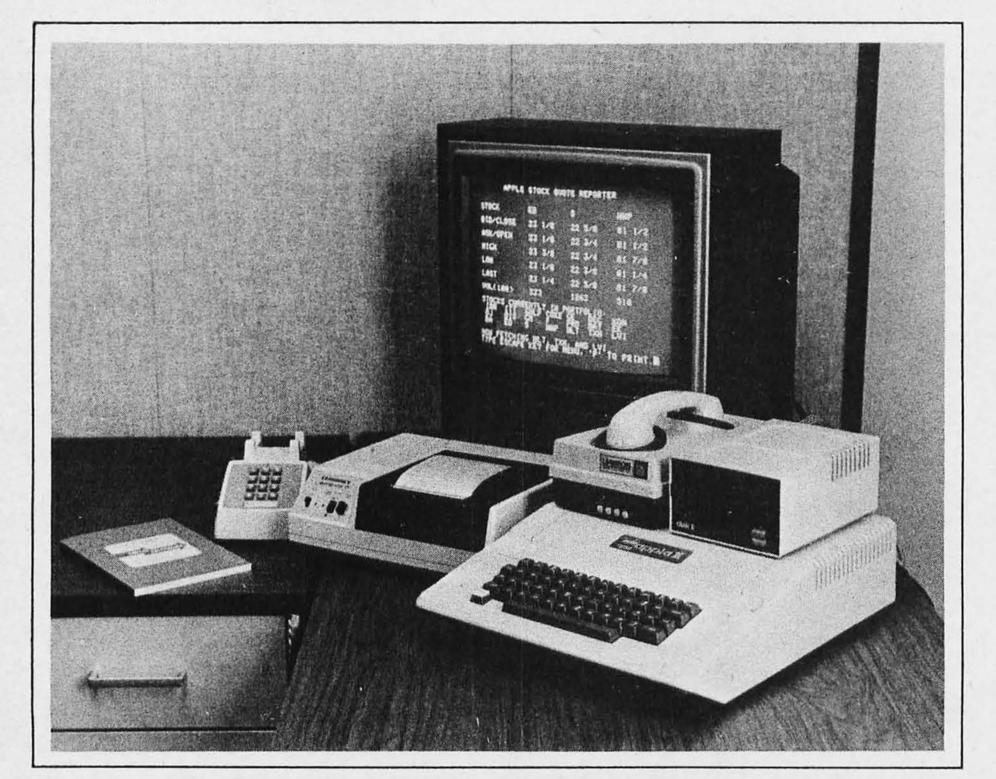
Circle 584 on inquiry card.

Run Program at 0000 with Your Poly 88

Having trouble running software beginning at hexadecimal 0000 on your PolyMorphic Poly 88 computer, due to the on board system monitor? A new Super-Monitor is now available which permits the use of low off board programmable memory for program storage. The Super-Monitor plugs into the two remaining read only memory sockets on the Poly processor card and leaves the original monitor intact. The monitor features: dump, move, verify, erasable read only memory (EROM), programmer, fill, in, out, cassette save, cassette load, search, serial port driver, GOTO, memory modify and return to Poly monitor. It is available as a set of two 2708 erasable read only memories with complete documentation for $59 plus shipping from Computer Hobbies Unlimited, 9215 Midlothian Tpk, Richmond VA 23235.■

Circle 585 on inquiry card.

Apple Users Get Access to Dow Jones Information Service



Apple Computer Inc has announced a new service to provide owners of its computers with stock portfolio information and other financial services.

Using a telephone linkup, users of Apple II computers will be able to dial Dow Jones' Stock Quote Reporter Service for 15 minute delayed stock and bond quotations. This information along with software provided by Apple will enable the user to determine current portfolio value, short and long term gains, and rate of return, among other things. At a later date, Apple II users will also be able to call up current news on companies in their user's portfolio.

The cost of the stock quote service includes a 1 time fee of $25 plus a usage charge of $3 for the first three minutes plus $.50 a minute thereafter for each usage session.

Contact Apple Computer Inc, 10260 Bandley Dr, Cupertino CA 95014.■

Circle 588 on inquiry card.

Cromemco Trace System Simulator

Trace is a system simulator program designed to facilitate assembly language program development on Cromemco computer systems. Trace emulates the behavior of a Z-80 processor as it follows the logic of the user program.

Virtually all aspects of system operation can be simulated, including prioritized interrupts and input/output (IO) commands. Trace options include control of register display and choice of display frequency. A historical record of program execution is maintained in a 100 instruction circular queue.

Features which locate errors quickly include: warnings if the user writes to unexpected areas, simulation of IO commands on the console, undefined calls to CDOS routines, improper return from subroutine calls, and execution of branch instructions or decimal adjust if the relevant flags are in an undefined state.

The advanced features of Trace enable it to be used in place of logic analyzers or in-circuit emulators in program development.

It is available on minifloppy (Model TSS-S) or floppy (Model TSS-L) IBM format disks for $95. For additional information contact Cromemco Inc, 280 Bernardo Av, Mountain View CA 94040.■

Circle 586 on inquiry card.

Software Packaged in North Star Format

MicroAge has introduced new applications software on minifloppy disk that is packaged in North Star format. Each of the programs is ready to run in any S-100 8080 or Z-80 computer system. They include: financial programs, mathematical analysis programs, statistical programs and miscellaneous programs, volume 1 and 2 games, backorder program, mailing list, and North Star disk operating system for the Centronics printer. Priced at $35 each, they are available from MicroAge Mail Order, 803 N Scottsdale Rd, Tempe AZ 85281.■

Circle 587 on inquiry card.

Simulation Package for Minicomputers

Mini-Dynamo is a simulation package designed for the PDP-11, NOVA, Eclipse, Varian and other minicomputers. Mini-Dynamo offers standard Dynamo equation formats, easily specified tabular and graphical output (with automatic or user specified scaling), automatic sorting of equations for correct computational order, ability to rerun the model without recompiling, and a variety of options to tailor output to users' needs (Dynamo is a continuous simulation language for large computers).

The one time license fee is $2500, or $1000 for educational institutions. Mini-Dynamo can be licensed from Pugh-Roberts Associates Inc, 5 Lee St, Cambridge MA 02139.■

Circle 589 on inquiry card.

ITHACA AUDIO

THE OEM MARKETPLACE

IA Expands S-100 Line

Video Display Board

Featuring a full 128 upper/lower case ASCII character set stored in a 1K buffer memory. Easy to read 16 line x 64 character format can be displayed on an inexpensive video monitor or a modified TV set. Includes a TTY software driver. Add our powerful K 2 FDOS to create a versatile operator console.

$25.00

Disk Controller Board

Controls up to 4 single or double sided drives. Data protect features include automatic disable of write-gate during power-down for data integrity. Supported by a reliable software package, K 2 FDOS and complete diagnostic documentation.

$35.00

K2 Operating System

Power full disk software in the DEC tradition. Includes Text Editor (TED), File Package (PIP), Debugger (HDT), Assembler (ASMBLE), HEXBIN, 1 COPY, System Generator (SYSGEN). Command syntax follows Digitals OS-8, RT/11 format. First in a family of high level software. Soon to be released, FORTRAN & Pascal Compilers.

$75.00

Field-proven reliable engineering

Over 10,000 boards worldwide prove Ithaca Audio provides the quality and reliability you demand.

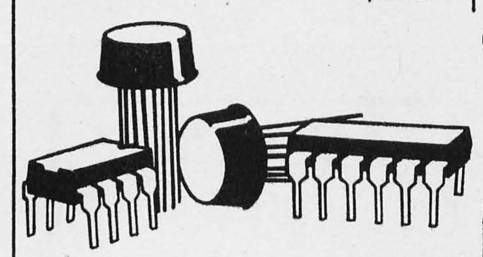
Ithaca Audio Boards are fully S-100 compatible, featuring gold edge connectors and plated-through holes. All boards (except the Protoboard) have fully buffered data and address lines, DIP switch addressing, solder mask and parts legend.

Z-80 CPU Board Most powerful 8 bit central processor available. Featuring power-on-jump, provision for on-board 2708. Accepts most 8080 software. **$35.00**

8K Static RAM Board High speed static memory at the lowest cost per bit. Includes memory protect/unprotect and selectable wait states. **$25.00**

2708/2716 EPROM Board Indispensable for storing dedicated programs and often used software. Accepts up to 16K of 2708's or 32K of 2716's. **$25.00**

Protoboard Universal wire-wrap board for developing custom circuitry. Accepts any size DIP socket. **$25.00**



RAM!

32K for $359.

Ithaca Audio is now stocking the Mostek 4115 add-on RAM for S.D.'s Expandoram. Buy their basic board, 32K of RAM from us and SAVE.

S.D. SALES Expandoram board	$199
Ithaca Audio 32 4115's @ $5.00 ea.	160
32K Only	**$359**

Mass Storage at Incomparable Prices.

Ithaca Audio Floppy Disk

- Up to 250K bytes, single sided
- Up to 500K bytes, double sided
- Data protect
- Powerful software operating system includes 8 utility programs, text editor.

Add the capacity of full size disk to your S-100 microcomputer. Controller, Disk Drive, and Software available separately.

Memorex single sided **550 Flexible Disk Drive**	**$456.**
Memorex double sided **552 Flexible Disk Drive**	**$630.**
Disk Controller Board	**$35.**
K2 FDOS Available on 8" floppy disk w. manual	**$75.**

Quality Components

ZILOG Z-80	$19.00
ZILOG Z-80A	23.00
INTEL 2708	11.00
FAIRCHILD 2102 LHPC	1.60
FAIRCHILD 2102 LIPC	1.35

IMSAI 8080 Kit with 22 Slot M.B.
$560.00
plus $10.00 shipping.

HOW TO ORDER

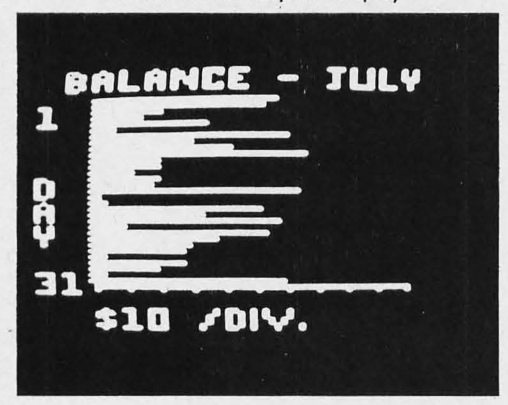
Send check or money order, include $2.00 shipping per order. N.Y.S. Residents include tax.

For technical assistance call or write to:

ITHACA AUDIO
P.O. Box 91
Ithaca, New York 14850
Phone: 607/273-3271

Circle 190 on inquiry card.

Graphics Software for 6800 Computers with a GT-61 Graphics Display



Graph #1 and Graph #2 are utility packages enabling use of the SwTPC GT-61 graphics display with a 6800 computer. Graph #1 is for use with a machine language, while Graph #2 is for use through SwTPC 8 K BASIC, version 2. When using Graph #2 the user can program the GT-61 display directly from a BASIC program without any machine language programming.

Both versions can display, as well as erase, points, lines and the complete upper case ASCII character set. There is additional capability for user defined characters. Applications for these packages include display of statistical data complete with legends, and games.

Each package is priced at $5.50 for paper tape or $6.50 for Kansas City standard cassette. The package includes a 22 page instruction manual with sample programs and a full source listing of Graph. To order, contact Applied Microcomputer Systems, POB 68, Silver Lake NH 03875.■

Circle 643 on inquiry card.

CP/M Available for Users of Micropolis Disk System

Micropolis disk users can now join the software bus. Without any hardware changes CP/M can be run with all the features available to users of the system on standard floppy disks. CP/M on Micropolis (version 1.4) is available for $145. The following is also available for the CP/M user with Micropolis hardware: Microsoft FORTRAN-80, $400; Microsoft disk extended BASIC, $300; CBASIC compiler/interpreter BASIC, $95; MAC macroassembler, $100; SID symbolic instruction debugger, $85; and BASIC-E compiler/interpreter BASIC, $30. For more information about the above products write to Lifeboat Associates, 164 W 83rd St, New York NY 10024.■

Circle 644 on inquiry card.

Business Software Package

A business software package has been announced by Aaron Associates, POB 1720A, Garden Grove CA 92640. The package includes a general ledger, accounts receivable, accounts payable and payroll package, inventory and manufacturing package, and a mailing list package.

Required equipment for this package includes a line printer (Okidata 22 preferred), terminal (Soroc IQ 120 preferred), dual North Star disk drive system with North Star BASIC and 32 K bytes of memory.

The package comes with ten named files, over 70 programs, a user's manual and workbook, and 200 pages of program listings. The price is $177.■

Circle 645 on inquiry card.

BASIC Business Software for 6800 Computers

Three new software programs for 6800 systems using BASIC are now available from the Stephen Moe Company, POB 595, Springfield OR 97477. The software is designed to run on the SwTPC processor with Smoke Signal, SwTPC or PerCom disks or cassettes.

The inventory software provides a capacity of up to 1000 80-character items per disk. Contents are item search, daily activity report, minimum quantity search, list by item, list by class, list by vendor, access a different file, update a file, and create a new file.

The payroll software tabulates payroll records, prints paychecks, lists employee records, summarizes employer tax records, creates new files, opens a different file and assembles end of year or quarter records.

The billing program prints mailing labels, bills, overdue and aging notices, as well as providing reports for customer accounts receivable, sales and last purchase. It also has file handling capabilities.

All programs run in 16 K bytes. They are available on disk or cassette for $200 each.■

Circle 646 on inquiry card.

Free PET Computer Services

Two free PET services are available through the Microcomputer Resource Center Inc, 5150 Anton Dr, Room 212, Madison WI 53719. They include the PET Cassette Exchange, a program library whereby free exchange programs are available for the PET computer on cassette with no service charge. The *Ultimate PET Resource Handbook* is a continually updated listing of all hardware and software sources for the PET. For a free copy of this handbook send a self-addressed stamped envelope.■

Circle 647 on inquiry card.

P.O. Box 4430X Santa Clara, CA 95054
For will call only: (408) 988-1640
2322 Walsh Ave.

Quest ELECTRONICS

Same day shipment. First line parts only. Factory tested. Guaranteed money back. Quality IC's and other components at factory prices.

QUEST Cosmac Super Elf Computer $106.95

Compare features before you decide to buy any other computer. There is no other computer on the market today that has all the desirable benefits of the **Super Elf** for so little money. The Super Elf is a small single board computer that does many **big** things. It is an excellent computer for training and for learning programming with its machine language and yet it is easily **expanded** with **additional memory, Tiny Basic, ASCII Keyboards, video character generation, etc.**

The **Super Elf** includes a **ROM** monitor for program loading, editing and execution with **SINGLE STEP for program debugging** which is not included in others at the same price. With **SINGLE STEP** you can see the microprocessor chip operating with the **unique Quest** address and data bus displays **before, during** and **after** executing instructions. Also, CPU mode and instruction cycle are shown on several LED indicator lamps.

An RCA 1861 **video graphics** chip allows you to connect to your own TV with an inexpensive video modulator to do graphics and games. There is a **speaker system** included for writing your own music or using many music programs already written. The speaker amplifier may also be used to drive relays for control purposes.

A 24 key **HEX keyboard** includes 16 HEX keys plus **load, reset, run, input, memory protect, memory select, monitor select** and **single step**. Large, on board displays provide output and optional **high and low address**. There is a 44 pin standard connector for PC cards and a 50 pin connector for the **Quest Super Expansion Board**. Power supply and sockets for **all** IC's are included in the price plus a detailed 90 page instruction manual.

Many schools and universities are using the Super Elf as a course of study. OEM's use it for training and research and development.

Remember, other computers only offer Super Elf features at additional cost or not at all. **Compare before you buy. Super Elf Kit $106.95, High address option $8.95, Low address option $9.95. Custom Hardwood Cabinet** with drilled and labelled front panel **$24.95. NiCad Battery Backup Kit $4.95.** All kits and options also come completely assembled and tested.

Questdata, a 12 page monthly software publication for 1802 computer users is available by subscription for $12.00 per year.

Tiny Basic for ANY 1802 System

Cassette $10.00. On ROM Monitor $38.00. Super Elf owners, 30% off. **Object code listing or paper tape with manual $5.50. Original ELF Kit Board $14.95.**

Super Expansion Board with Cassette Interface $89.95

This is truly an astounding value! This board has been designed to allow **you** to decide how you want it optioned. **The Super Expansion Board comes with 4K of low power RAM** fully addressable anywhere in 64K with built-in memory protect and a **cassette interface**. Provisions have been made for all other options on the same board and it fits neatly into the hardwood cabinet alongside the **Super Elf**. The board includes slots for up to 6K of **EPROM** (2708, 2758, 2716 or TI 2716) and is **fully socketed** ($12.00 value). EPROM can be used for the monitor and Tiny Basic or other purposes.

A IK Super ROM Monitor $19.95 is available as an on board option in 2708 EPROM which has been preprogrammed with a program loader/editor and error checking multi file cassette read/write software, (relocatible cassette file) another exclusive from Quest. It includes register save and readout, video graphics driver with blinking cursor and block move capability. The Super Monitor is written with subroutines allowing users to take advantage of monitor functions simply by calling them up. Improvements and revisions are easily done with the monitor. If you have the **Super Expansion Board** and **Super Monitor** the monitor is up and running at the push of a button.

Other on board options include **Parallel Input and Output Ports** with **full handshake**. They allow easy connection of an ASCII keyboard to the input port. **RS 232 and 20 ma Current Loop** for teletype or other device are on board and if you need more memory there are two **S-100** slots for static RAM or video boards. A Godbout 8K RAM board is available for $127.95. **Parallel I/O Ports $9.85, RS 232 $4.50, TTY 20 ma I/F $1.95, S-100 $4.50. A 50 pin connector set** with ribbon cable is available at $12.50 for easy connection between the **Super Elf** and the **Super Expansion Board.**

The **Power Supply** for the Super Expansion Board is a 5 amp supply with + 8v ± 18v + 12v − 5v. Regulated voltages are ±5v & +12v $29.95. −12 volt optional. Deluxe version includes the case at $39.95.

INTEGRATED CIRCUITS

7400TTL	
7400N	.17
7402N	.17
7404N	.19
7409N	.23
7410N	.17
7414N	.63
7420N	.17
7422N	1.39
7430N	.20
7442N	.50
7445N	.69
7447N	.60
7448N	.69
7450N	.17
7474N	.29
7475N	.49
7485N	.88
7489N	2.00
7490N	.43
7492N	.43
7493N	.43
7495N	.69
74100N	.90
74107N	.29
74121N	.34
74123N	.59
74125N	.39
74145N	.69
74150N	.95
74151N	.69
74154N	1.00
74157N	.69
74161N	.87
74162N	.87
74163N	.87
74174N	.96
74175N	.90
74190N	1.15
74192N	.87
74193N	.85
74221N	1.55
74298N	1.65
74365N	.66
74366N	.66
74367N	.66
74LS00 TTL	
74LS00N	.25
74LS02N	.25
74LS04N	.25
74LS05N	.25
74LS08N	.25
74LS10N	.25
74LS13N	.40
74LS14N	.90
74LS20N	.25
74LS22N	.25
74LS28N	.41
74LS30N	.25
74LS33N	.39
74LS38N	.30
74LS74N	.35
74LS75N	.47
74LS90N	.51
74LS93N	.51
74LS95N	1.89
74LS107N	.35
74LS112N	.35
74LS113N	.35
74LS132N	.72
74LS136N	.35
74LS151N	.67
74LS155N	.67
74LS157N	.67
74LS162N	.91
74LS163N	.91
74LS174N	.95
74LS190N	1.06
74LS221N	1.95
74LS258N	.67
74LS367N	.89
LINEAR	
CA3045	.90
CA3046	.67
CA3081	1.80
CA3082	1.90
CA3089	2.95
LM301AN/AH	.35
LM305H	.87
LM307N	.35
LM308N	.89
LM309H	1.15
LM309K	.95
LM311H/N	.90
LM317T/K	2.92
LM318	1.35
LM320K-5	1.20
LM323K-5	6.95
LM320K-12	1.35
LM320K-15	1.35
LM320T-5	1.60
LM320T-8	1.60
LM320T-12	1.50
LM320T-15	1.60
LM324N	1.15
LM339N	1.55
LM340K-5	1.10
LM340K-8	1.10
LM340K-12	1.10
LM340K-15	1.10
LM340K-24	1.10
LM340T-5	1.10
LM340T-8	1.10
LM340T-12	1.10
LM340T-15	1.10

LM340T-18	1.10
LM340T-24	1.10
LM343H	4.50
LM370	1.15
LM377	4.50
LM379	5.00
LM380N	1.00
LM381	1.60
LM382	1.60
LM703H	.40
LM709H	.28
LM723H/N	.50
LM733N	.67
LM741CH	.35
LM741N	.25
LM747H/N	.62
LM748N	.35
LM1303N	.82
LM1304	1.10
LM1305	1.27
LM1307	2.00
LM1310	2.75
LM1458	.47
LM1800	1.75
LM1812	7.50
LM1889	3.00
LM2111	1.75
LM2902	1.50
LM3900N	.60
LM3905	1.75
LM3909N	.61
MC1458V	.50
NE540L	2.89
NE550N	.65
NE555V	.43
NE556A	.79
NE565A	1.00
NE566V	1.50
NE567V	1.20
NE570B	5.00
NE571B	5.00
78L05	.60
78L08	.60
79L05	.70
78M05	.85
75108	1.75
75491CN	.50
75492CN	.55
75494CN	.89
A to D CONVERTER	
8038B	4.50
8700CJ	13.95
8701CN	22.00
8750CJ	13.95
LD130	9.95
9400CJV/F	7.40
ICL7103	9.50
ICL7107	14.25
CMOS	
CD34001 Fair.	.50
CD4000	.16
CD4001	.21
CD4002	.21
CD4006	1.10
CD4007	.21
CD4008	.21
CD4009	.39
CD4010	.39
CD4011	.21
CD4012	.21
CD4013	.36
CD4014	.86
CD4015	.86
CD4016	.36
CD4017	.94
CD4018	.94
CD4019	.21
CD4020	1.02
CD4021	1.02
CD4022	.86
CD4023	.21
CD4024	.75
CD4025	.21
CD4026	1.51
CD4027	.36
CD4028	.79
CD4029	1.02
CD4030	.21
CD4035	1.02
CD4040	1.02
CD4042	.71
CD4043	.63
CD4044	.63
CD4046	1.67
CD4049	.36
CD4050	.36
CD4051	1.13
CD4060	1.42
CD4066	.71
CD4068	.40
CD4069	.40
CD4070	.40
CD4071	.21
CD4072	.21
CD4073	.21
CD4075	.21
CD4076	1.75
CD4078	.40
CD4081	.21
CD4082	.21
CD4116	.47
CD4490	5.50
CD4507	1.00
CD4508	4.25
CD4510	1.02

CD4511	.94
CD4515	2.52
CD4516	1.10
CD4518	1.02
CD4520	1.02
CD4527	1.51
CD4528	.79
CD4553	5.75
CD4566	2.25
CD4583	4.50
CD4585	1.10
CD40192	3.00
74C00	.28
74C04	.33
74C10	.28
74C14	2.10
74C20	.28
74C30	.28
74C48	1.95
74C74	.75
74C76	1.40
74C90	1.15
74C93	1.40
74C154	3.00
74C160	1.44
74C175	1.35
74C192	1.65
74C221	2.00
74C905	3.00
74C906	.75
74C914	1.95
74C922	5.50
74C923	5.50
74C925	6.95
74C926	6.95
74C927	6.95
INTERFACE	
8095	.65
8096	.65
8097	.65
8098	.65
8T09	1.25
8T10	4.50
8T13	3.00
8T20	5.50
8T23	3.10
8T24	3.50
8T25	3.20
8T26	1.69
8T28	2.75
8T97	1.69
8T98	1.69
MOS/MEMORY RAM	
2101-1	3.95
2102-1	.95
2102AL-4	1.60
21F02	1.25
2104A-4	4.95
2107B	4.95
2111-1	3.75
2112-2	3.95
2114	8.50
4116	17.95
2513B	6.30
21L02-1	1.49
MM5262	.40
MM5280	3.00
MM5320	9.95
MM5330	5.94
PD411D-3	4.00
PD411D-4	5.00
P5101L	13.95
4200A	9.95
82S25	2.90
91L02A	1.75
HD0165-5	6.95
MM57100	4.50
GIAY38500-1	9.95
MCM6571A	9.95
9368	3.50
410D	10.00
416	16.00
CLOCKS	
MM5309	3.00
MM5311	3.60
MM5312	4.80
MM5313	3.60
MM5314	3.90
MM5315	4.00
MM5316	5.00
MM5318	3.60
MM5369	2.10
MM5841	14.45
MM5865	7.95
CT7001	5.80
CT7002	8.95
CT7010	8.95
CT7015	7.25
MM5375AA/N	3.90
MM5375AB/N	4.90
7205	16.50
7207	7.50
7208	15.95
7209	4.95
DS0026CN	3.75
DS0056CN	3.75
MM53104	2.50
MICROPROCESSOR	
6800	17.50
6802	24.95
8080A with data	8.95
8085	27.00
Z80	29.95

8212	2.90
8214	8.00
8216	2.90
8224	2.90
8228	5.35
8251	8.50
8253	10.00
8255	9.25
8257	19.50
8259	19.50
CDP1802CD	19.95
CDP1802D	25.00
CDP1861	12.95
6820	9.95
6850	12.95
6502	12.50

IC SOCKETS

Solder Tin PIN	1 UP	Low Profile PIN	1UP
8	.15	22	.30
14	.18	24	.35
16	.20	28	.42
18	.27	36	.58
20	.29	40	.57

3 level wire wrap gold.
14 pin .25 16 pin .27
2 level 14 pin ww .20

WIRE WRAP LEVEL 3

PIN		PIN	
14	.25	24	.86
16	.27	28	1.00
18	.57	40	1.23

UART/FIFO	
AY5-1013	5.50
AY5-1014	7.50
3341	6.95
PROM	
1702A	3.95
N82S23	2.95
N82S123	3.50
N82S126	3.75
N82S129	3.75
N82S131	3.75
N82S136	8.75
N82S137	8.75
2708	12.50
DM8577	2.90
8223	2.90
2716T1	22.50
2716 Intel	48.00
CRYSTALS	
1 MHz	4.50
2 MHz	4.50
4 MHz	4.25
5 MHz	4.25
10 MHz	4.25
18 MHz	3.90
20 MHz	3.90
32 MHz	3.90
32768 MHz	4.00
1.8432 MHz	4.50
3.5795 MHz	1.20
2.0100 MHz	1.95
2.097152 MHz	4.50
2.4576 MHz	4.50
3.2768 MHz	4.50
5.0688 MHz	4.50
5.185 MHz	4.50
5.7143 MHz	4.50
6.5536 MHz	4.50
14.31818 MHz	4.25
18.432 MHz	4.50
22.1184 MHz	4.50
CONNECTORS	
44 pin edge	2.00
100 pin edge	4.50
100 pin edge WW	4.50
KEYBOARD ENCODERS	
AY5-2376	$12.50
AY5-3600	13.50
74C922	5.50
74C923	5.50
HD0165-5	8.95

IC Test Clips

	1	10
Red	.55	.47
Black	.55	.47

Keyer 8043 14.50
comp. w/spac./sock.

TRANSISTORS	
2N1893	.40
2N2222A	.18
2N2369	.30
2N2904A	.20
2N2907A	.25
2N3053	.40
2N3638	.25
2N3643	.25
2N3904	.18
2N3906	.18
2N3055	.69
2N4400	.25
2N4401	.75
2N4402	.20
TIP31	.60
TIP33A	1.00

CLOCK MODULES Complete alarm clocks ready to hook up with transformer and switches. Very compact with .50" and .84" digits.

MA1002A, C or E .50"	**8.95**
102P3 Transformer	**2.25**
MA1010A, C or E .84"	**11.95**
102P2 Transformer	**2.25**
Special transformer and six switches when purchased w/module	**2.95**
MA1003 car module .3" green fluor. display	**15.95**

RESISTORS ¼ watt 5%
10 per type .03 1000 per type .012
25 per type .025 350 piece pack
100 per type .015 5 per type 6.75

KEYBOARDS	
56 key ASCII keyboard kit	$62.50
Fully assembled	72.50
53 key ASCII keyboard kit	55.00
Fully assembled 65.00 Enclosure	14.95
LEDS	
Red T018	.15
Green, Orange, Yellow T018	.20
Jumbo Red	.20
Green, Orange, Yellow Jumbo	.25
Cliplite LED Mounting Clips	8/$1.25

(specify red, amber, green, yellow, clear)

CONTINENTAL SPECIALTIES in stock
Complete line of breadboard test equip.
MAX-100 8 digit Freq. Ctr. $128.95

OK WIRE WRAP TOOLS in stock
Portable Multimeter $18.00

DIGITAL THERMOMETER $48.50
Batt. oper. General purpose or medical 32°–230°F. Disposable probe cover ±.2° accuracy. Comp. Assy. in compact case.

COMPUTER BOARD KITS	
8K RAM Board Kit	$134.95
4K EPROM Kit	114.95
I/O Board Kit	44.50
Extender Board w/connector	12.50
Video Interface board kit	125.00
16K EPROM board kit w/o PROMS	74.50
16K Static RAM board kit	395.00
North Star Floppy Disk Kit	$665.00
Additional Drive Kit	415.00
SPECIAL PRODUCTS	
MM5865 Stopwatch Timer	9.00
PC board	7.50
Switches Mom. Pushbutton	.27
3 pos. slide	.25
Encoder HD0165-5	6.95
3 Digit Universal Counter Board Kit Operates 5–18 Volt DC to 5 MHz typ. .125" LED display	**10.50**
Voice actuated switch	.50
Paratronics 100A Logic Analyzer Kit	**$224.00**
Model 10 Trigger Expander Kit	$229.00
Model 150 Bus Grabber Kit	$369.00
TRANSFORMERS	
6V 300 ma	3.25
12 Volt 300 ma transformer	1.25
12.6V CT 600 ma	3.75
12V 250 ma wall plug	2.95
12V CT 250 ma wall plug	3.50
24V CT 400 ma	3.95
10V 1.2 amp wall plug	4.85
18V 6 amp	12.95

DISPLAY LEDS

MAN1	CA	.270	2.90
MAN3	CC	.125	.39
MAN72/74	CA/CA	.300	1.00
DL704	CC	.300	1.25
DL707/DL707R	CA	.300	1.00
DL727/728	CA/CC	.500	1.90
DL747/750	CA/CC	.600	1.95
DL750	CC	.600	1.95
FND359	CC	.357	.70
FND500/507	CC/CA	.500	1.35
FND503/510	CC/CA	.500	.90
FND800/807	CC/CA	.800	2.20
3 digit Bubble			.60
4 digit bubble			.80
DG8 Fluorescent			1.75
DG10 Fluorescent			1.75
5 digit 14 pin display			1.00
NSN69 9 digit display			.60
7520 Clairex photocells			.39
TIL311 Hex			9.50

COMPUTER GRADE CAPS

1600	mfd	200V	4.50
2000	mfd	45V	2.50
3200		50V	2.50
5500		25V	2.50
5800		40V	3.00
6100		40V	3.00
6100		50V	3.00
7700		50V	2.50
8000		55V	2.50
9400		65V	3.00
10000		20V	2.50
12000		30V	3.50
35000		25V	5.00
55000		25V	5.00
82000		15V	6.00
160000		10V	6.00

Auto Clock Kit $15.95

DC clock with 4-.50" displays. Uses National MA-1012 module with alarm option. Includes light dimmer, crystal timebase PC boards. Fully regulated, comp. instructs. Add $3.95 for beautiful dark gray case. Best value anywhere.

RCA Cosmac VIP Kit 229.00

Video computer with games and graphics.

Not a Cheap Clock Kit $14.95

Includes everything except case. 2-PC boards. 6-.50" LED Displays. 5314 clock chip, transformer, all components and full instrucs. Green and orange displays also avail. **Same kit w/.80" displays.** Red only. **$21.95**

60 Hz Crystal Time Base Kit $4.40

Converts digital clocks from AC line frequency to crystal time base. Outstanding accuracy. Kit includes: PC board, IC, crystal, resistors, capacitors and trimmer.

Digital Temperature Meter Kit

Indoor and outdoor. Switches back and forth. Beautiful. 50" LED readouts. Nothing like it available. Needs no additional parts for complete, full operation. Will measure −100° to +200°F, tenths of a degree, air or liquid. Very accurate. **$39.95**
Beautiful woodgrain case w/bezel **$11.75**

NiCad Battery Fixer/Charger Kit

Opens shorted cells that won't hold a charge and then charges them up, all in one kit w/full parts and instructions. **$7.25**

PROM Eraser

Ultraviolet, assembled **$34.50**

Clock Calendar Kit $23.95

CT7015 direct drive chip displays date and time on .6" LEDS with AM-PM indicator. Alarm/doze feature includes buzzer. Complete with all parts, power supply and instructions, less case.

Sinclair 3½ Digit Multimeter

Batt./AC oper. 1mV and .1NA resolution. Resistance to 20 meg. 1% accuracy. Small, portable, completely assem. in case. 1 yr. guarantee. Best value ever! **$59.95**

Video Modulator Kit $8.95

Convert your TV set into a high quality monitor without affecting normal usage. Complete kit with full instructions.

2.5 MHz Frequency Counter Kit

Complete kit less case **$37.50**

30 MHz Frequency Counter Kit

Complete kit less case **$47.75**
Prescaler kit to 350 MHz **$19.95**

1978 IC Update Master Manual

1978 IC Update Master Manual **$19.50.** Complete IC data selector, 2175 pg. Master reference guide. Over 42,000 cross references. Free update service through 1978. Domestic postage **$3.50.** Foreign **$5.00.** 1979 IC Update available late Jan. **$30.00.**

Stopwatch Kit $26.95

Full six digit battery operated. 2–5 volts. 3.2768 MHz crystal accuracy. Times to 59 min., 59 sec., 99 1/100 sec. Times std., split and Taylor. 7205 chip, all components minus case. Full instructions.

D Connectors RS232

25 Pin Subminiatures

DB25P	2.95	DE9P	1.50
DB25S	3.95	DE9S	1.95
Cover	1.50	DA15P	2.10
RS232 Complete Set	6.50	DA15S	3.10

S-100 Computer Boards

8K Static RAM Kit	$127.00
16K Static RAM Kit	265.00
24K Static RAM Kit	423.00
32K Dynamic RAM Kit	449.00
64K Dynamic RAM Kit	945.00
8K/16K Eprom Kit (less PROMS)	**$89.00**
Video Interface Kit	**$139.00**
Motherboard **$39.** Extender Board	**$8.99**

**TERMS: $5.00 min. order U.S. Funds. Calif residents add 6% tax.
BankAmericard and Master Charge accepted.
Shipping charges will be added on charge cards.**

FREE: Send for your copy of our NEW 1978 QUEST CATALOG. Include 28¢ stamp.

Circle 311 on inquiry card.

Unclassified Ads

FOR SALE: TVC video board from Digital Group, $75. Heath IO-12 scope, $45. Super CT-1024 with monitor, $299. SwTPC PR-40 printer, $195. 4 K memory boards for 6800, $69. 8 K Seals memory for 6800, $169. Smoke Signal disc for 6800, $639. All prices firm. Will consider trades. Stuart Brown, 1116B Easton Av, Somerset NJ 08873, (201) 249-7972.

FOR SALE: Boris Chess Computer, complete with board, pieces and instruction manual. Decision time for moves from 5 seconds to 99 hours. Plays Black/White. Standard algebraic chess notation will display current position of pieces on request; with 2 minutes decision time, beats micro-1.5 in 17 to 30 moves. Cost $300 new, will sell for $200. Rick Racine, 445 S 72nd Kansas City KS 66111.

FOR SALE: Meca dual cassette and controller, digital systems dual 8 inch and controller, 2 each. 16 K TDL memory boards. Make offer. (714) 770-7789.

FOR SALE: New 32 K Dynamic Digital Group memory, factory assembled. Discovered that my new software requires static. $675 (I pay postage). Frank Fitch, 2347 A Market Street, San Francisco CA 94114, (415) 543-6345 work or (415) 861-4881 home.

TRS-80 OWNERS: I am interested in surveying TRS-80 users. Write Prof Bill Parks, Chase-203, State University College, 1300 Elmwood Av, Buffalo NY 14222.

FOR SALE: Microprocessor computer program course and trainer computer, All built and in perfect condition, $275 shipping included. N Swan, 4839 Beaune Rd, Ludington MI 49431.

FOR SALE: Altair 8800, with 16 slot mother board plus 16 edge connectors, cooling fan modification, 2SIO-serial IO with both ports, 88-PIO parallel IO board, 1 K static memory with 512 bytes of memory, 12 K static memory, 4 K programmable memory software board with 8080 assembler, text editor, system monitor, and all documentation. All assembled and tested $1000, you pay shipping. Don Cheeseman, 8231 Creekline Dr, San Antonio TX 78251, (512) 681-4938.

FOR SALE: DIGI-LOG Telecomputer II portable terminal with 5 inch monitor and carrying case. ASCII code, built-in acoustic coupler. Suitable for timesharing computer applications. Will ship UPS, $1200. Gene Witherup, RD #4, Bloomsburg PA 17815, (717) 784-5366.

FOR SALE: (in Canada) IMSAI 8080 with 22 slots, $1000. 32 K of TDL 250 ns memory, $1100. ADM3 display, only $1000. Teletype model 40 3001 per minute printer, $2800. ICOM dual disk full size floppy, $2400. Prefer to sell system complete with interface cards, cables, FDOS3 operating system, Assembler, BASIC (DEBBI), and editor, $8500. Ron Cox, 2208 Victoria Av, Windsor Ontario, CANADA, N8X 1R1 (519) 969-9692.

NEW UNCLASSIFIED POLICY

Readers who have equipment, software or other items to buy, sell or swap should send in a clearly typed notice to that effect. To be considered for publication, an advertisement must be clearly noncommercial, typed double spaced on plain white paper, contain 75 words or less, and include complete name and address information.

These notices are free of charge and will be printed one time only on a space available basis. Notices can be accepted from individuals or bona fide computer users clubs only. We can engage in no correspondence on these and your confirmation of placement is appearance in an issue of BYTE.

Please note that it may take three or four months for an ad to appear in the magazine.■

WANTED: Manual for Instant Input Assembler-Version 1 (for 9900 Super Starter System by Technico). Will pay at least original cost. William T Wilson, 203 Oxford St, Portland ME 04111.

FOR SALE: Remex high-speed punched paper tape reader and spooler model RR-101 D/R. Requires a parallel interface. Asking $50 and you pay shipping. Charles B Wall, Rt #3, Clarksville TN 37040, (615) 552-2199.

FOR SALE OR TRADE: Technico Super Starter System, assembled and working. Uses TMS 9900 16 bit processor. Has 32 IO lines, serial port, 2 K programmable memory, programmable read only memory monitor/assembler, erasable read only memory programmer, etc. Includes power supply, $450. Also REMEX high-speed papertape reader, $125. Trade for plotter or color monitor. Jim McCord, 330 Vereda Leyenda, Goleta CA 93017, (805) 968-6681.

FOR SALE: Szerlip "programmable memory setter," erasable read only memory programmer for S-100 bus computers. Programs 1702A, 2708 erasable read only memories — can also program 2716s. Assembled version, supplied with all cabling, documentation and software. Never used. Current list price $375, first offer over $200 accepted. Also, 16 INTEL 1702A erasable read only memories, $3 each, lot of 16 only. Edwin J Kroeker, 46 Woodcliff Rd, Newton MA 02161, (617) 527-6369.

FOR SALE: For LSI-11, Heath H-11 owners. 4 K memory core — use as small disk, keep loader or BASIC resident in memory. Also 16 K dual width Intel memory. $425 each, $700 for both. Ed Judge (413) 584-7159 anytime.

FOR SALE: Cromemco Z2-D Microcomputer, 21 slots, blue table cabinet, 48 K programmable memory, one minifloppy, Dynabyte naked terminal, all 4 MHz compatible, with all cables in running condition with 12 inch monitor, U/C ASCII bps, 16 K BASIC, CDOS, editor, etc, and games in BASIC. Vincent Pinto, Gate Hill Rd, Stony Point NY 10980, (914) 947-2740.

FOR SALE: Cromemco A-D Converter, $180. Factory built, like new. A Bob, 148 W77, New York NY 10024.

FOR SALE: Complete Xitan system. Includes Z-processor; SMB 1; 32 K of programmable memory (two Z-16s); TDL video display board; TDL 12 K BASIC in read only memory and on cassette tape (relocatable); 16 K read only memory board; interface 1 board which provides I/O (standard EIA 25 pin connectors) for RS-232 and 20MA. All documentation and tapes necessary to operate system. Also Digital Group keyboard. Asking $2400. Terry Young, 4 Aiken St, Derry NH 03038.

FOR SALE: New Slo-Syn Model MO91-FD-318 200 steps per revolution stepping motors. Operate on 5.9 VDC. In original cartons with literature and switching order. Intended for servo mechanisms but can be used for other things. $20. D C Suits, 2619 Essex Rd, Ann Arbor MI 48104.

FOR SALE: TDL Xitan system: Z-80 Z-processor board, Z16 16 K memory, and SMB interface board; 8 K Godbout memory; all standard TDL software (including 12 K Super BASIC) on cassette. Runs perfectly. Original cost over $1900, first certified check for $1300 takes it. Also MERLIN S-100 graphics board with 1 K read only memory monitor and 320 X 200 resolution, $300. Bert Katzung, 65 Knoll Rd, San Rafael CA 94901, (415) 456-5812.

FOR SALE: PERTEC iCOM FDOS-II dual drive/single density floppy disk drive with Intel interface card. Almost new—paid $3,300—will take best offer received one month after offer appears. Gary Miner, POB 1177, Santa Cruz CA 95061, (408) 429-1331.

WANTED: January 1976 BYTE. Will buy or trade. Have October, November, December 1976 for trade or $1 each. Glenn Whitham, 9 Trinity Pl, Wayland MA 01778.

FOR SALE: Heathkit H8 and H9, two cassette recorders, fully assembled with 8 K of memory. Included in package are Heath BASIC, assembler, editor, and debug manuals. Selling for $1175, almost $200 less that original unassembled kit price. Contact Jon Nareff, 366 Great Swamp Rd, Glastonbury CT (203) 633-2060.

FOR SALE: Challenger 2 processor with 8 K polled keyboard and 540 video board, $650. Bruce Higer, 19 Woodland Dr, Sands Pt NY 11050, (516) 883-9543.

FOR SALE: For Selectric IO 731 series, 24/48 V power supply, data and power supply cables and plugs, plus full documentation and software on interface to S-100 bus. First $50 takes it. Daniel Eisenberg, 1507 Sharon Rd, Tallahassee FL 32303, (904) 385-4148.

FOR SALE: Complete system. Altair 8800b with 48 K programmable memory, SOROC video terminal, two North Star disks, Diablo RO printer. All assembled and operating. Will sell all or part. Many extras. Send SSAE for details and prices. G Briggs, 6195 Deer Path, Manassas VA 22110.

FOR SALE: L/N, TI59, PC100A printer, 40 new program cards, master library read only memory, static read only memory, static packette, manuals. Reason for sale: data analysis for research project finished. First money order for $320 gets all. A Brandwein, 745 Fifth Av, New York NY 10022, (212) P15-6081.

FOR SALE: Rockwell XPO-1 development system for PPS-4/1 microprocessors with optional assemblers, power supply and 1 K programmable memory. Never used. $450. G Hyman, 22 Cross Hill Av, Yonkers NY 10703, (914) 476-2129.

FOR SALE: Two National Multiplex 3M3-A digital cassette recorders and 2S10R interface/monitor board. Storage capacity per 3M data cartridge of up to 2 M bytes. Search capability up to 50 k. Will sell for best price over $350. B Wagman, POB 57091, Washington DC 20037.

FOR SALE: Heath H11 computer (LSI-11 based) with 16 KB programmable memory, serial IO board, parallel IO board, H10 paper tape reader/punch, EIS/FIS extended instruction chip, documentation, software. 100 percent assembled and tested. $1800 or best offer. Dave Morrill, 1260 NW 17 Av #4, Rochester MN 55901, (507) 282-0758 evenings.

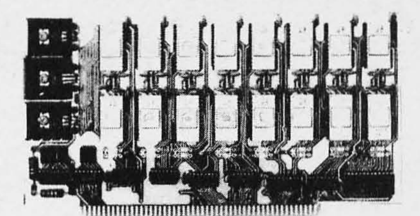
FOR SALE: Complete set of BYTE magazines, volume 1, # 1 through volume 3, # 8 (August 1978). Excellent condition. Make offer. M A Richards, 1470 Hampton Glen Ct, Decatur GA 30033.

SCHOOL COMPUTER NEEDS HELP: In need of driver software, instructions or documentation for interfacing a Monroe cassette recorder with a Monroe 8080 for program storage. Also information on mark sense card hookup. I'm new to computing and need it for a school computer. Robert Geier, 4875 Countryside, Lyndhurst OH 44124.

FOR SALE: Heath H-9 video Terminal. Up and running, all documentation. $425 and I pay shipping. Andy Thornburg, RR2, Thompsonville IL 62890, (618) 627-2166.

FOR SALE: DEC PDP-8m, mint, with teletype interface, with 4 K core memory, $1100. With 8 K core memory $1800. 8 K MOS board $475. Most peripherals and software available, repairs and custom interfacing, systems consulting, trades. PORTACOM, briefcase, impact, ASCII terminals complete with modems. Reconditioned with manuals $595. New cost is $1500. New boxed ASR33, $1100. Used, $595. K2DCY, POB 632, W Caldwell, NJ 07006. (201) 226-9185 evenings.

16K EPROM CARD-S 100 BUSS



$59.95 KIT

OUR BEST SELLING KIT!

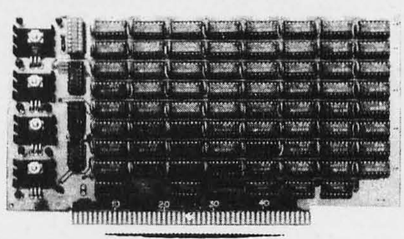
USES 2708's!

Thousands of personal and business systems around the world use this board with complete satisfaction. Puts 16K of software on line at **ALL TIMES!** Kit features a top quality soldermasked and silk-screened PC board and first run parts and sockets. All parts (except 2708's) are included. Any number of EPROM locations may be disabled to avoid any memory conflicts. Fully buffered and has WAIT STATE capabilities.

OUR 450NS 2708'S ARE $8.95 EA. WITH PURCHASE OF KIT

ASSEMBLED AND FULLY TESTED ADD $25

8K LOW POWER RAM KIT-S 100 BUSS



$129 KIT

Use 21L02 450 NS RAMS!

Thousands of computer systems rely on this rugged, work horse, RAM board. Designed for error-free, NO HASSLE, systems use.

KIT FEATURES:

1. Doubled sided PC Board with solder mask and silk screen layout. Gold plated contact fingers.
2. All sockets included.
3. Fully buffered on all address and data lines.
4. Phantom is jumper selectable to pin 67.
5. FOUR 7805 regulators are provided on card.

Blank PC Board w/Documentation **$29.95**
Low Profile Socket Set...**13.50**
Support IC's (TTL & Regulators) **$9.75**
Bypass CAP's (Disc & Tantalums) **$4.50**

ASSEMBLED AND FULLY BURNED IN ADD $30

16K STATIC RAM KIT-S 100 BUSS

$295 KIT



FULLY STATIC, AT DYNAMIC PRICES

WHY THE 2114 RAM CHIP?

We feel the 2114 will be the next industry standard RAM chip (like the 2102 was). This means price, availability, and quality will all be good! Next, the 2114 is FULLY STATIC! We feel this is the **ONLY** way to go on the S-100 Buss! We've all heard the HORROR stories about some Dynamic Ram Boards having trouble with DMA and FLOPPY DISC DRIVES. Who needs these kinds of problems? And finally, even among other 4K Static RAM's the 2114 stands out! Not all 4K static Rams are created equal! Some of the other 4K's have **clocked** chip enable lines and various timing windows **just as critical** as Dynamic RAM's. Some of our competitor's 16K boards use these "tricky" devices. But not us! The 2114 is the ONLY logical choice for a trouble-free, straightforward design.

KIT FEATURES

1. Addressable as four separate 4K Blocks
2. ON BOARD BANK SELECT circuitry (Cromemco Standard!). Allows up to 512K on line!
3. Uses 2114 (450NS) 4K Static Rams
4. ON BOARD SELECTABLE WAIT STATES
5. Double sided PC Board, with solder mask and silk screened layout. Gold plated contact fingers.
6. All address and data lines fully buffered
7. Kit includes ALL parts and sockets
8. PHANTOM is jumpered to PIN 67.
9. LOW POWER: under 2 amps TYPICAL from the +8 Volt Buss.
10. Blank PC Board can be populated as any multiple of 4K

BLANK PC BOARD W/DATA—**$33**
LOW PROFILE SOCKET SET—**$12**
SUPPORT IC'S & CAPS—**$19.95**
ASSEMBLED & TESTED—**ADD $30**
2114 RAM'S—**8 FOR $69.95**

16K STATIC RAM KIT SWTPC (SS-50) 6800 BUSS

USES 2114 4K RAMS!

$295 COMPLETE KIT

WHY PAY MORE FOR FINNICKY DYNAMIC BOARDS?

At last an affordable static RAM board for this popular buss. Quality PC Board with solder mask and silk-screen. Fully buffered with plenty of bypassing for reliable operation. FOUR ON-BOARD REGULATORS.

16K DYNAMIC RAM CHIP

16K X 1 Bits. 16 Pin Package. Same as Mostek 4116-4. 250 NS access. 410 NS cycle time. Our best price yet for this state of the art RAM. 32K and 64K RAM boards using this chip are readily available. These are new, fully guaranteed devices by a major mfg. **VERY LIMITED STOCK!**

8 FOR $89.95

NOT ASSOCIATED WITH DIGITAL RESEARCH OF CALIFORNIA, THE SUPPLIERS OF CPM SOFTWARE.

2708 EPROMS

450 NS!

Now full speed! Prime new units from a major U.S. Mfg. 450 N.S. Access time. 1K x 8. Equiv. to 4-1702 A's in one package.

~~$15.75 ea.~~ **$9.95** ~~4 FOR $50.00~~

PRICE CUT

NATIONAL SEMICONDUCTOR

JUMBO CLOCK MODULE

MA1008A BRAND NEW!



$6.95 (2 FOR $13)

(AC XFMR $1.95)

FEATURES
- FOUR JUMBO ½ INCH LED DISPLAYS
- 12 HR REAL TIME FORMAT
- 24 HR ALARM SIGNAL OUTPUT
- 50 OR 60 Hz OPERATION
- LED BRIGHTNESS CONTROL
- POWER FAILURE INDICATOR
- SLEEP & SNOOZE TIMERS
- DIRECT LED DRIVE (LOW RFI)
- COMES WITH FULL DATA

ASSEMBLED! NOT A KIT!

ZULU VERSION!
We have a limited number of the 24 HR Real time version of this module in stock.
#MA1008D — $9.95

PERFECT FOR USE WITH A TIMEBASE.

COMPARE AT UP TO TWICE OUR PRICE!

MANUFACTURER'S CLOSEOUT!

WESTERN DIGITAL UART

TR1602A. PIN FOR PIN SUB FOR AY5-1013 AND TMS6011.
FOR SERIAL I/O

$2.99 EACH

SURPLUS SPECIAL

SALE!
1N4148 DIODES. SILICON.
Same as 1N914. New, factory prime, Full Leads.
100 FOR $2
1000 FOR $17.50

New! REAL TIME Computer Clock Chip
N.S. MM5313. Features BOTH 7 segment and BCD outputs. 28 Pin DIP. **$4.95 with Data**

Z-80 PROGRAMMING MANUAL

By MOSTEK, or ZILOG. The most detailed explanation ever on the working of the Z-80 CPU CHIPS. At least one full page on each of the 158 Z-80 instructions. A MUST reference manual for any user of the Z-80. 300 pages. Just off the press.

$12.95

8212 I/O CHIP

8 BITS. 8080 or Z-80 SUPPORT CHIP. 24 PIN CHIP. NEW BY N.E.C.

SPECIAL: $2.25 EACH

MOTOROLA 7805R VOLTAGE REGULATOR. Same as standard 7805 except 750 MA output. TO-220. 5VDC output.

44¢ each or 10 for $3.95

Digital Research Corporation
(OF TEXAS)
P. O. BOX 401247Y GARLAND, TEXAS 75040 • (214) 271-2461

TERMS: Add 30¢ postage, we pay balance. Orders under $15 add 75¢ handling. No C.O.D. We accept Visa, MasterCharge and American Express cards. Tex. Res. add 5% Tax. Foreign orders (except Canada add 20% P & H. 90 Day Money Back Guarantee on all items.

Circle 100 on inquiry card.

Reader Service

To get further information on the products advertised in BYTE, fill out the reader service card with your name and address. Then circle the appropriate numbers for the advertisers you select from the list. Add a 15 cent stamp to the card, then drop it in the mail. Not only do you gain information, but our advertisers are encouraged to use the marketplace provided by BYTE. This helps us bring you a bigger BYTE.

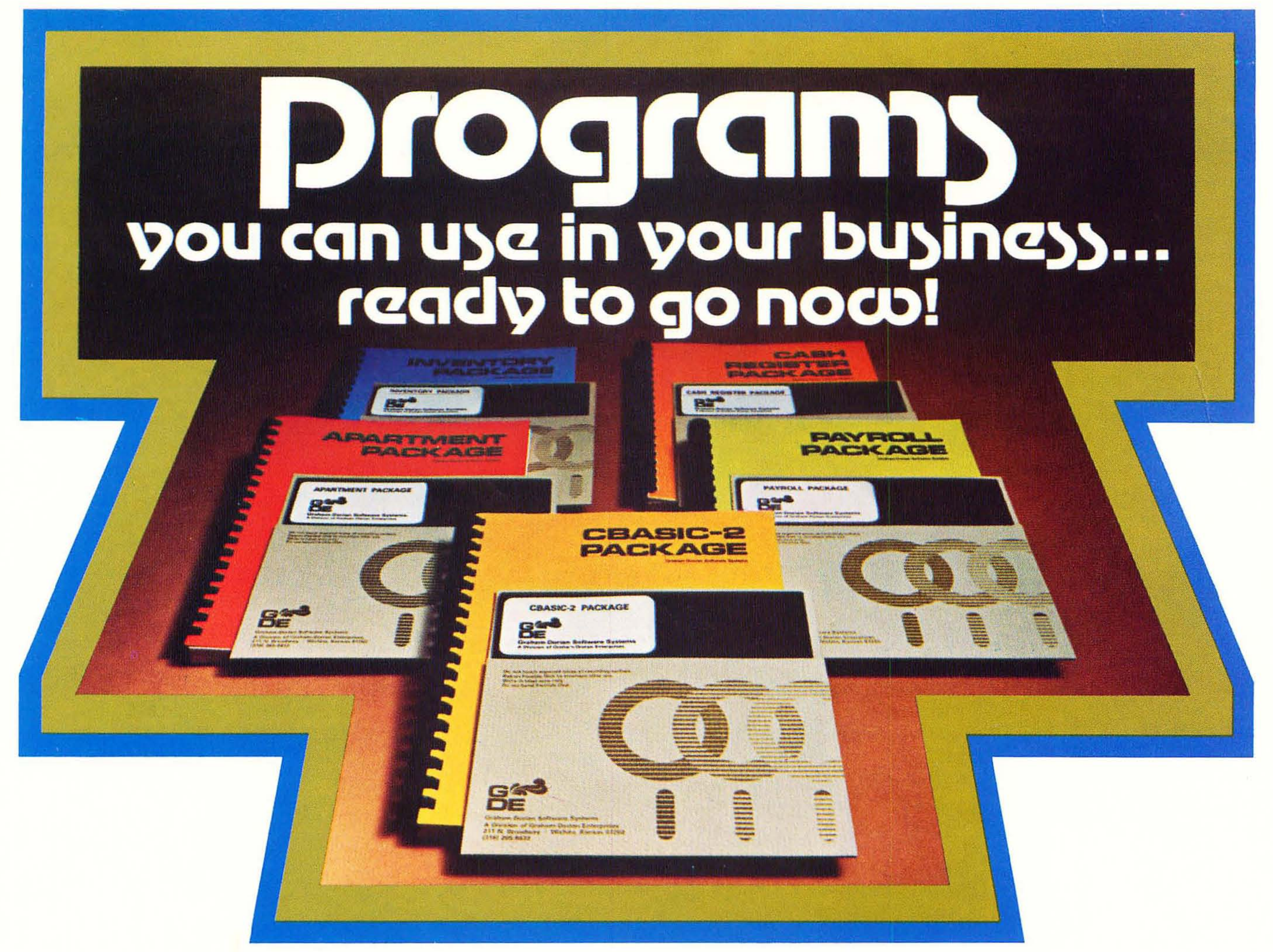
**Correspond directly with Company*

BOMB— BYTE's Ongoing Monitor Box

A Chess Blitz in the October BOMB

The first three articles in the October BOMB were all chess related. The first prize of $100 goes to Dan and Kathe Spracklen for their article "First Steps in Computer Chess Programming," page 86. The $50 second prize goes to Peter W Frey and Larry R Atkin for "Creating a Chess Player: An Essay on Human and Computer Chess Skill," page 182. Placing third was "A Computer Chess Tutorial," page 168, by Norman D Whaland, and "A Tiny Pascal Compiler, Part 2," page 34, by Kin-Man Chung and Herbert Yuen finished fourth.

Readers who wish to cast votes for their favorite articles in this month's BYTE may use the BOMB card provided on the opposite page.■



Our programs will let you realize the full potential of your hardware.

We developed these programs because we needed them in our businesses, and, try as we might, could not purchase them. They're on-line now, working for us and others around the country.

As users ourselves we know the problems from your perspective — not just as a manufacturer of software. The bugs are out and they're ready now to go to work helping make your life easier, keeping you in better control of your business.

Our first four program packages are: • **Apartment Management • Cash Register • Inventory • Payroll**

Here's a typical program

To give you an idea of the thoroughness of these programs, here's a summary of what the inventory package does for you. Gives a detailed listing of items in inventory and itemizes all goods sold from inventory, including which sales person sold what, when it sold and for how much ... recaps on one sheet this same inventory activity information ... investigates and changes any information in inventory, on request ... prints list of items to be re-ordered ... provides profit analysis comparing sales personnel and/or various products. And it can be inter-connected with our cash register package as well, for total program management.

Each of our initial programs is conceived, proven and offered with this same exacting thoroughness and attention to detail.

We stay with you after the sale

We're in this for the long haul and our support program is dedicated to that objective. Registered program owners receive:

• Periodic newsletters which include users' ideas and information exchange, plus tips to owners on further increasing benefits of the package through updated operational flexibility.

• Availability of software technicians to provide immediate answers to questions, via phone or mail.

• Customer rewrites and adaptations available on request, at added cost.

CBASIC-2 free

It takes the world's most powerful commercial basic to run our programs and we deliver it to you free.

Each of our program packages contains a disk with CBASIC-2 Compiler, CBASIC-2 Run Command and your Graham-Dorian software programs in INT and BAS file form. You also receive User's Manuals and Hard Copy Source Listing. At a price which pays for itself!

CBASIC-2 was developed and written by Software Systems, the people who wrote CBASIC, and includes many powerful enhancements.* All systems are compatible with any Z-80 or 8080 CP/M(TM) system. They are deliverable in standard eight-inch disk — either double or single density — or mini-floppy disk.

Give us a call or fill out the Reader Service Card in this issue. We promise a response within 24 hours of receipt. That's the kind of information service we expect, and know you do too.

*CBASIC-2 may be purchased separately from Graham-Dorian Software Systems for $89.95.

Master Charge and Visa cards accepted

GDE

Graham-Dorian Software Systems
A Division of Graham-Dorian Enterprises
211 N. Broadway / Wichita, Ks. 67202 / (316) 265-8633

Circle 153 on inquiry card.

Considering a Microcomputer?

Be Sure to Check Out the Product Offerings of the World's Largest Full Line Microcomputer Company.

All Ohio Scientific machines come with microcomputing's fastest full feature BASIC-in-ROM or on-Disk for instant use.

Challenger I Series

Economical computer systems that talk in BASIC. Ideal for hobbyists, students, education and the home.

	Minimum Configuration	Base Price
Superboard II — World's first complete system on a board including keyboard, video display, audio cassette, BASIC-in-ROM and up to 8K RAM	4K RAM	$ 279
Challenger IP — Fully packaged Superboard II with power supply	4K RAM	$ 349
Challenger IP Disk — Complete mini-floppy system expandable to 32K RAM	16K RAM	$1190

Challenger IIP Series

Ultra high performance BUS oriented microcomputers for personal, educational, research and small business use.

	Minimum Configuration	Base Price
C2-4P — The professional portable	4K RAM	$ 598
C2-8P — The world's most expandable personal machine for business or research applications	4K RAM	$ 799
C2-4P Disk — The ultimate portable	16K RAM	$1464
C2-8P Single Disk — Ideal for education, advanced personal users, etc.	16K RAM	$1738
C2-8P Dual Disk — Most cost effective small business system	32K RAM	$2597

Challenger II Serial Interface Series

Same great features as Challenger IIP Series for those who have serial terminals: small business, education, industry.

	Minimum Configuration	Base Price
C2-0—Great starter for users with a terminal	4K RAM	$ 298
C2-1 — Great timeshare user accessory; cuts costs by running simple BASIC programs locally	4K RAM	$ 498
C2-8S — Highly expandable serial machine, can add disks, etc.	4K RAM	$ 545

Challenger III The Ultimate in Small Computers

The unique three processor system for demanding business, education, research and industrial development applications.

	Minimum Configuration	Base Price
C3-S1 — World's most popular 8″ floppy based microcomputer	32K RAM dual floppys	$3590
C3-OEM — Single package high volume user version of C3-S1	32K RAM dual floppys	$3590
C3-A — Rack mounted multi-user business system directly expandabe to C3-B	48K RAM dual floppys	$5090
C3-B — 74 million byte Winchester disk based system. World's most powerful microcomputer	48K RAM dual floppys	$11,090

OHIO SCIENTIFIC also offers you the broadest line of expansion accessories and the largest selection of affordable software!

Compare the closest Ohio Scientific Model to any other unit you are considering. Compare the performance, real expansion ability, software and price, and you will see why we have become the world's largest full line microcomputer company.







I'm interested in OSI Computers. Send me information on:

- ☐ Personal Computers
- ☐ Small Business Computers
- ☐ Educational Systems
- ☐ Industrial Development Systems
- ☐ I'm enclosing $1.00 for your 64-page small computer buyer's guide.

Ohio residents add 4% tax.

Name ______________________

Address ______________________

City ______________________

State ______________ Zip ________

Phone ______________________

OHIO SCIENTIFIC
1333 S. Chillicothe Road
Aurora, Ohio 44202
(216) 562-3101

Circle 290 on inquiry card.